

**Yalova Üniversitesi**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**Web Programlama Final Ödevi**

**Hazırlayanlar**

**Dilan Durmaz 160101021**

**Seda Yılmaz 160101057**

**ARALIK-2019**

İçindekiler

[**PROJE NASIL ÇALIŞTIRILIR?** 3](#_Toc27629106)

[**HEADER.Vue** 3](#_Toc27629107)

[**FOOTER.VUE** 6](#_Toc27629108)

[**KAYIT.VUE** 6](#_Toc27629109)

[**GIRISYAP.VUE** 9](#_Toc27629110)

[**FİLTREKAHVE.VUE** 11](#_Toc27629111)

[**MENU.VUE** 12](#_Toc27629112)

[**SLIDER.VUE** 13](#_Toc27629113)

[**ATOLYE.VUE** 14](#_Toc27629114)

[**BLOG.VUE** 15](#_Toc27629115)

[**HAKKIMIZDA.VUE** 16](#_Toc27629116)

[**KUTUCİKOLATA.VUE** 17](#_Toc27629117)

[**MAGAZALAR.VUE** 18](#_Toc27629118)

[**ÜRÜNBİLGİ.VUE** 19](#_Toc27629119)

[**ÜYELİK.VUE** 20](#_Toc27629120)

[**BROWSER TAB DEĞİŞİMİ** 22](#_Toc27629121)

[**VUE-LOADİNG** 23](#_Toc27629122)

[**ROUTER KULLANIMI** 23](#_Toc27629123)

[**KAHVE.VUE** 25](#_Toc27629124)

[**CİKOLATA.VUE** 26](#_Toc27629125)

[**AKSESUAR.VUE** 28](#_Toc27629126)

[**GIFT YAPIMI** 29](#_Toc27629127)

[**ARAMA BUTONU POPUP** 31](#_Toc27629128)

[**PRE-LOADER YAPIMI** 33](#_Toc27629129)

[**TEKLİ VE ÇOKLU LİSTELEME** 34](#_Toc27629130)

[**SERVER OLUŞTURMA** 35](#_Toc27629131)

[**İNDEX.JS** 35](#_Toc27629132)

[**DATA.JSON** 35](#_Toc27629133)

[**UYELER.JSON** 36](#_Toc27629134)

[**HOME.JS** 36](#_Toc27629135)

[**UYE.JS** 37](#_Toc27629136)

**PROJE NASIL ÇALIŞTIRILIR?**

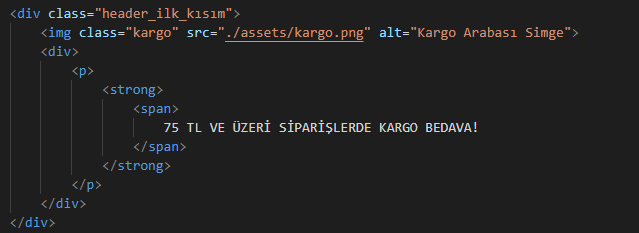
İlk olarak Visual Studio’da iki ayrı pencerede server ve wb dosyaları açılır. Server dosyasını çalıştırabilmek için terminale node index.js komutu yazılır.

Wb dosyası için ise yine terminal açılır,npm run serve yazarak proje başlatılır.

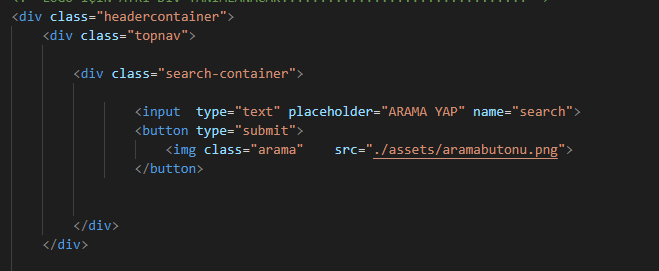
**HEADER.Vue**

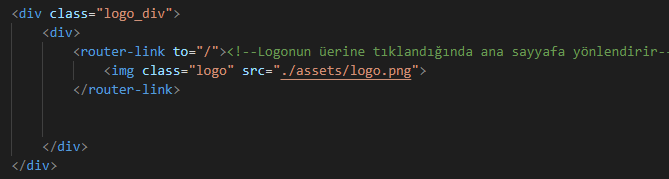
**1.HTML Kodları**

İlk olarak orijinal sayfa bulunan görsel ve yazılar <img/> ve <p/> tagleri ile eklendi.<span/> tag’i ile satır içi gruplandırma yapıldı. <strong/> tag’i ile güçlendirilecek metin tanımlandı. <img/> tag’i içinde görselin kaynağı tanımlanırken assets alanına ulaşabilmek için görseldeki gibi tanımlama yapıldı.

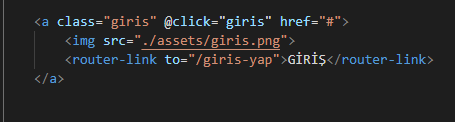


Arama butonu tanımlaması yapıldı. <input/> etiketi ile kullanıcının veri girişi yapabilmesi için giriş alanı tanımlandı.

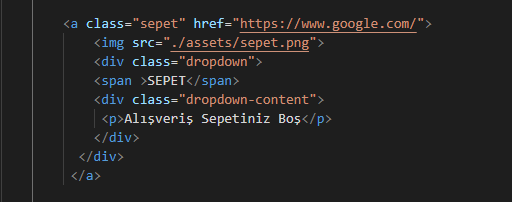




Aktiflik sağlayıp sayfa geçişi yapılacak görsele click özelliği eklendi.Yine sayfa yönlendirmesi yapılacağından dolayı <router-link/> kullanıldı.

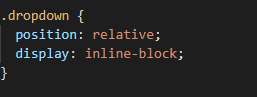


Sepet görselinin üzerine gelindiğinde sepetin boş olduğunu belirtmek için dropdown-content kullanıldı.



**2.CSS Kodları**

Display:inline-block özelliği ile satırda blok özelliği sağladık.Bu özellik genelde birden fazla elementi yanyana hizalamak için kullanılır.Bir elemana yükseklik özelliği tanımlayabilmek için onun blok olması , aynı satırda olmaları içinde inline olması gerekiyor.

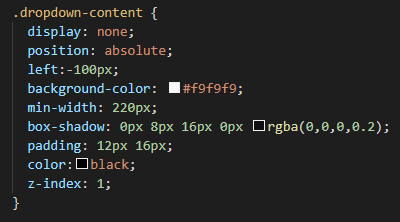


Z-index : Bir elementi z ekseni üzerinde üste ya da alta konumlandırmak için kullanılır.

Display:none özelliği , oluşturulan elemanın gözükmemesi için kullanılır.

Box-shadow : Gölge efekti için kullanılır.

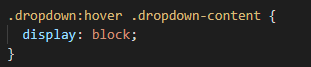
Padding : İçeriğin iç kısmında boşluk oluşturmak için kullanılır.



Dropdown üzerine gelindiğinde dropdown-content'in gözükmesi için display özelliği block olarak değiştirildi.

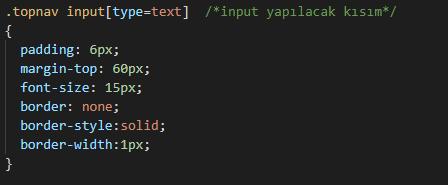
Cursor:pointer , eklendiği elemanın üzerine gelindiğinde fare imlecinin el işareti olmasını sağlar.

Float:left , ard arda sıralanmış elementlere float uygulandığında , elementler yeterli alan varsa yanyana dizilirler.Burada elementin sola dayalı olması sağlanmıştır.

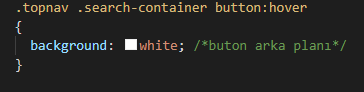


İnput yapılacak alanın css kodları aşağıda gösterilmiştir.Input type kısmında giriş alanının görüntülenme türü belirtilmiştir.

Border-style: solid ,border’ın kenarlarının nasıl stilde olacağı belirtilmiştir.



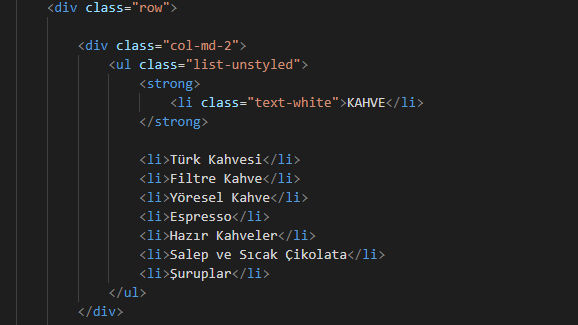
Arama kısmındaki butonun üzerine gelindiğinde(hover) background’un renginin değişmesi sağlandı.



# **FOOTER.VUE**

**1.Html Kodları**

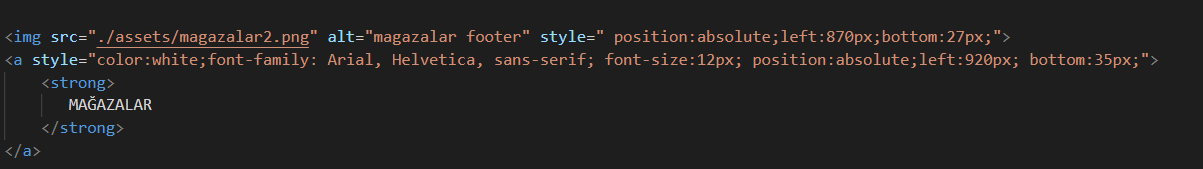
Sayfa düzeni sağlanması açısından satır ve sütun özellikleri kullanıldı.<ul/> etiketi ile sırasız bir liste tanımlandı. Liste öğelerini oluşturabilmek için de <li/> etiketi kullanıldı.



**2.CSS Kodları**

Eklenen görselin style özellikleri bu alanda tanımlandı.

Href içerisinde linkler tanımlanır.



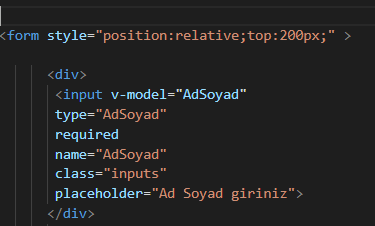
# **KAYIT.VUE**

**1.Html Kodları**

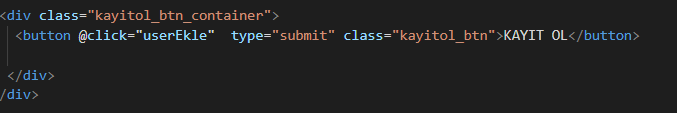
“Zaten hesabınız var mı ?” yazısına tıklandığında üye girişi sayfasına yönlendirmek için gerekli olan path tanımlandı. (“/uye.girisi”)



V-model ile two-way binding özelliği sağlandı. Binding özelliği bir objeyi diğer obje ile ilişkilendirmek için kullanılır. Yani objelerden biri değiştiğinde diğeri de otomatik olarak değişir. V-model form elemanları ile birlikte kullanılır.



KAYIT OL butonuna script’te yazılmış olan userEkle metodu @click=”userEkle” şeklinde çağırılır.



**2.Script Kodları**

Axios promise tabanlı olarak yayınlanmış,tarayıcı ve node.js tarafında çalışılabilir bir http clienttir.Terminale ;

npm install axios

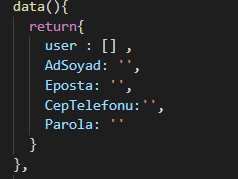
komutu yazılarak projeye indirilir. Aşağıdaki şekilde de import edilir.

import axios from "axios";

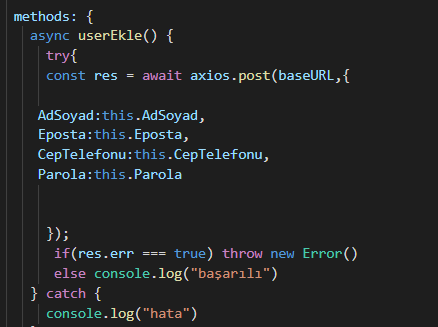
Ardından Server’da üyelere ait bilgilerin bulunduğu sayfanın url’si tanımlanır.

const baseURL ="http://localhost:7000/uye/kaydet"

user[] adında bir liste oluşturulur ve burada kullanıcının kayıt olurken girmesi gereken bilgiler bulunur.

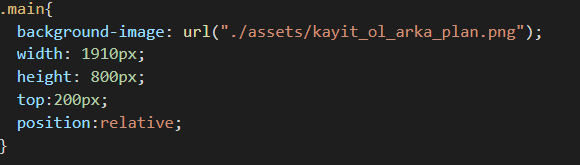
****

Burada userEkle metodu tanımlanmıştır. Girilen bilgiler burada tutulup, json’a gönderilir.Giderse başarılı döndürülür yanlış ise hata döner.Yukarıda belirtilen baseURL ile buradaki üyelere yeni üyeler eklenir.

****

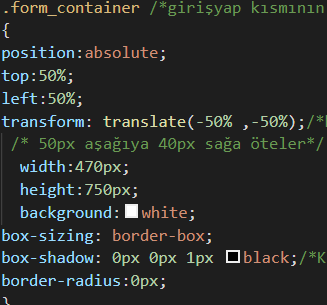
**3.Css Kodları**

Background-image: url ile arkaplana görsel eklendi.



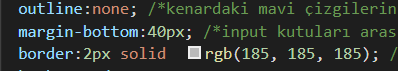
Transform:translate() , html elemanlarını döndürmek,boyutlandırmak,eğmek için kullanılır.Burada 50px aşağıya 40

Px sola ötelenmiş.

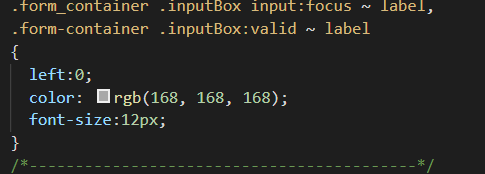


Outline: none ile elemanın kenarındaki mavi çizgiler ortadan kaldırıldı.

Border:2px ile border kalınlığı belirlendi.



Focus ile elemanın durum değişikliği kontrol edildi.Fare ile eleman üzerine gelindiğinde özellik değişir fakat geri çekildiğinde eski haline döner.



# **GIRISYAP.VUE**

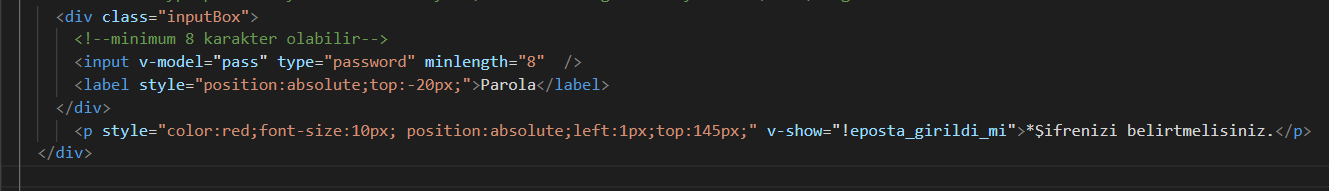
**1.Html Kodları**

Öncelikle giriş sayfası için <img/> etiketleri ile görseller eklendi.src içerisinde görselin yolu tanımlandı.href içerisinde gidilmesi gereken sayfanın bağlantı linki belirtildi.

V-model ile two-way binding özelliği sağlandı.Binding özelliği bir objeyi diğer obje ile ilişkilendirmek için kullanılır.Yani objelerden biri değiştiğinde diğeri de otomatik olarak değişir.V-model form elemanları ile birlikte kullanılır.

V-show data içerisindeki veri true ise bileşeni gösterir.Aksi takdirde false değeri ise göstermez.

Girilecek olan şifrenin maksimum uzunluğu minlength ile belirtilmiştir.



**2.Script Kodları**

Döndürülecek değerler data içerisinde tanımlanır(gerekirse data içerisinde varsayılan değerler tanımlanabilir).

Methods kısmında ise gerekli kontrollerin yapılacağı fonksiyon(eposta\_uyari) tanımlandı .Eğer verilen datalardan herhangi birinin değeri boş ise eposta\_girildi\_mi datası false olarak değişir. Ve uyarı mesajı alınır.

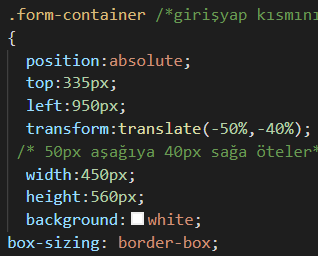


**3.Css Kodları**

Transform:translate() , html elemanlarını döndürmek,boyutlandırmak,eğmek için kullanılır.Burada 50px aşağıya 40

Px sola ötelenmiş.

Transition:.5s ile geçiş efekti özelliği verildi. Yani eleman üzerindeki değişikliklerin geçişliliğinin süresi için kullanılır.



# **FİLTREKAHVE.VUE**

Server’da bulunan filtre kahveler v-for döngüsü ile getirilir. Burada row ve col ile satır sütün ayrılıp 1 satır 4 sutün şeklinde veriler çekilir.

<br> etiketi ile alt satıra geçililir.

<strong> etiketi ile yazı kalınlaştırılır.****

FiltreKahveler adında liste oluşturulur.Girilen url fetch metodu ile getirilir.Buradaki işlem ile filtre adı altındaki ürünler çekilir ve listelenir.

Console.log ile filtre kahveler yazdırılır.

data:() =>({

       FiltreKahveler:[]

    }),

    mounted() {

        fetch(API\_URL)

        .then(response=>response.json())

        .then(result=> {

            this.FiltreKahveler=result.Kahve;

            console.log(this.FiltreKahveler)

        } )

    }

# **MENU.VUE**

Dropdown menü yapılmıştır.Yazıların üzerine gelinince menü aşağı doğru açılır.Kahve yazısına tıklanıldığında router-link sayesinde belirtilen sayfaya gidilmektedir.

 <router-link    to="/kahve-sayfasi">

Sayfalar arasına loading konulmuştur.Terminale yazılan

npm install --save vue-sweetalert-icons

komutu ile projeye dahil edilip,aşağıdaki şekilde kullanılmıştır.Bu sayede sayfalar arasında loading gifti döner.

<sweetalert-icon icon="loading" /> </p>

Gift, modal içinde çıkmaktadır.Boostrap kodu kullanılmıştır.Aşağıdaki şekilde düzenlemesi yapılmıştır.Modalın sayfada gerekli olmayan footer ve header kısımları true yapılıp kapatılmıştır.

<b-modal id="modal-1" hide-footer="true" hide-header="true" size="sm" body-bg-variant="light" title="BootstrapVue">

Script bölümünde ise openModel metodu tanımlanmıştır.Bu metod sayesinde modal açılır ve 1 saniye boyunca ekranda kalır.Bu değer arttırılıp veya azaltılabilir.setTimeout ile 1 saniye sonunda modal saklanır.Bu metod @click ile butona yazılır.

 openModal(){

            this.$bvModal.show('modal-1')

            var v = this;

            setTimeout(function(){

                v.$bvModal.hide('modal-1')

            },1000)

        }

# **SLIDER.VUE**

<b-carousel

      id="carousel-1"

      class="slaytkonum"

      v-model="slide"

      :interval="4000"

      controls

      indicators

      background="#ababab"

      style="text-shadow: 1px 1px 2px #333;"

      @sliding-start="onSlideStart"

      @sliding-end="onSlideEnd"

    >

      <!-- Text slides with image -->

      <b-carousel-slide>

            <template v-slot:img>

          <img

            class="d-block img-fluid w-100"

            src="../assets/kasık.png"

             width="1300"

            height="900"

            alt="image slot"

          >

        </template>

      </b-carousel-slide>

Slider’da b-carousel kullanılmıştır. İnterval’da slayttan slayta geçiş süresi belirtilmiştir. Src’de görselin konumu belirtilmiştir.width ile slaytın genişliği, height ile yüksekliği belirtilmiştir.

export default {

    data() {

      return {

        swiperOption: {

          slidesPerView: 3,

          spaceBetween: 30,

          pagination: {

            el: '.swiper-pagination',

            clickable: true

slidePerView’de slaytta bulunan görsel adedi belirtilmiştir.Görseller arası mesafe de 30 olarak ayarlanmıştır.

**ATOLYE.VUE**

İmg , p , h taglerinin yanısıra iframe tag’i kullanıldı.

Iframe nedir?

Ifareme ; sayfa içine doküman,video veya medya eklememizi sağlar.

Atöyle sayfası için, iframe yapısı kullanılarak sayfaya Youtube’dan video eklendi.

 <b-button style="position:relative;top:-400px;left:710px;z-index:999;" @click="modalShow = !modalShow"></b-button>

      <img style="position:relative;left:0px;top:-400px;width:1400px;height:600px;"  src="../assets/asdf.png" alt="Kahve resmi anasayfa">

    <b-modal hide-footer="true"  v-model="modalShow">

        <mdb-container>

    <div class="embed-responsive embed-responsive-21by9">

      <iframe class="embed-responsive-item"  src="https://www.youtube.com/embed/FtbRJQXgrOs" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>   ">

    </div>

  </mdb-container>

    </b-modal>

B-modal kullanarak ,butona basıldığında video popup şeklnde ekrana getirildi.

<a  href="https://www.biletix.com/etkinlik-grup/108904001/TURKIYE/tr">

    <img style="position:relative;left:0px;top:-300px;width:300px;height:500px;" src="../assets/d1.png" alt="Kahve resmi anasayfa">

    </a>

    <img style="position:relative;left:40px;top:-300px;width:300px;height:500px;"  src="../assets/d2.png" alt="Kahve resmi anasayfa">

    <button v-b-modal.my-modal style="color:black;background-color:white;border:none;outline:none;position:relative;left:-174px;top:-90px;">DETAYLI BİLGİ</button>

     <b-modal  hide-footer="true"  id="my-modal">



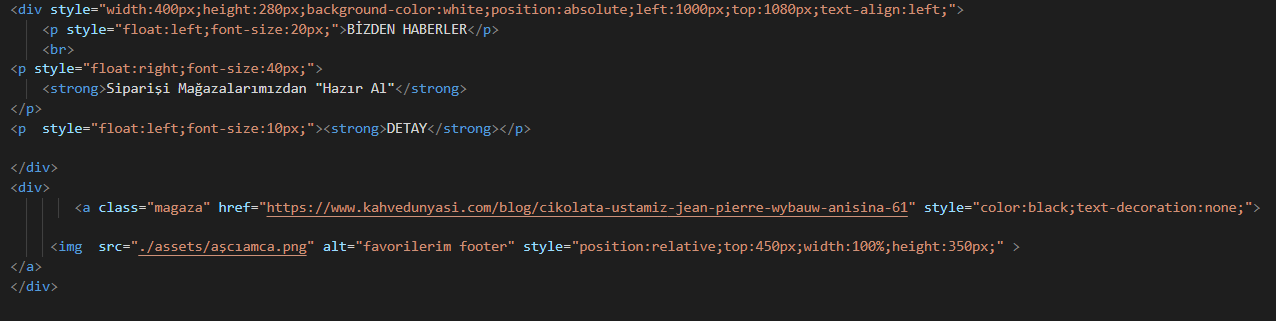
# **BLOG.VUE**

img tag’i ile görsel eklendi.

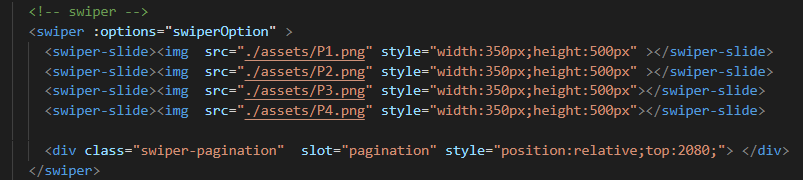
P tag’i ile paragraflar tanımlandı.

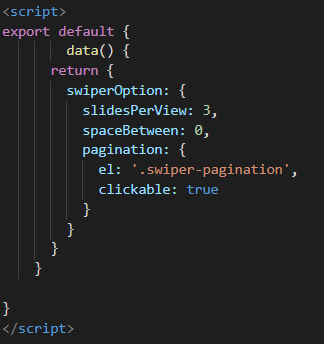
Görsellere link özelliği verildi.

Footer kısmında gözükecek alanların tanımı yapıldı.



Slider ile görsel geçişleri sağlandı.



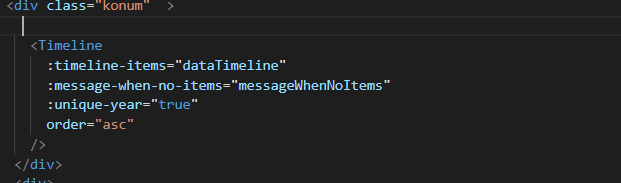


SlidesPerVie:3 , 3 adet görselin yanyana gözükmesini sağlar.Değeri arttırıldığında ekranda o kadar görsel yanyana durur.

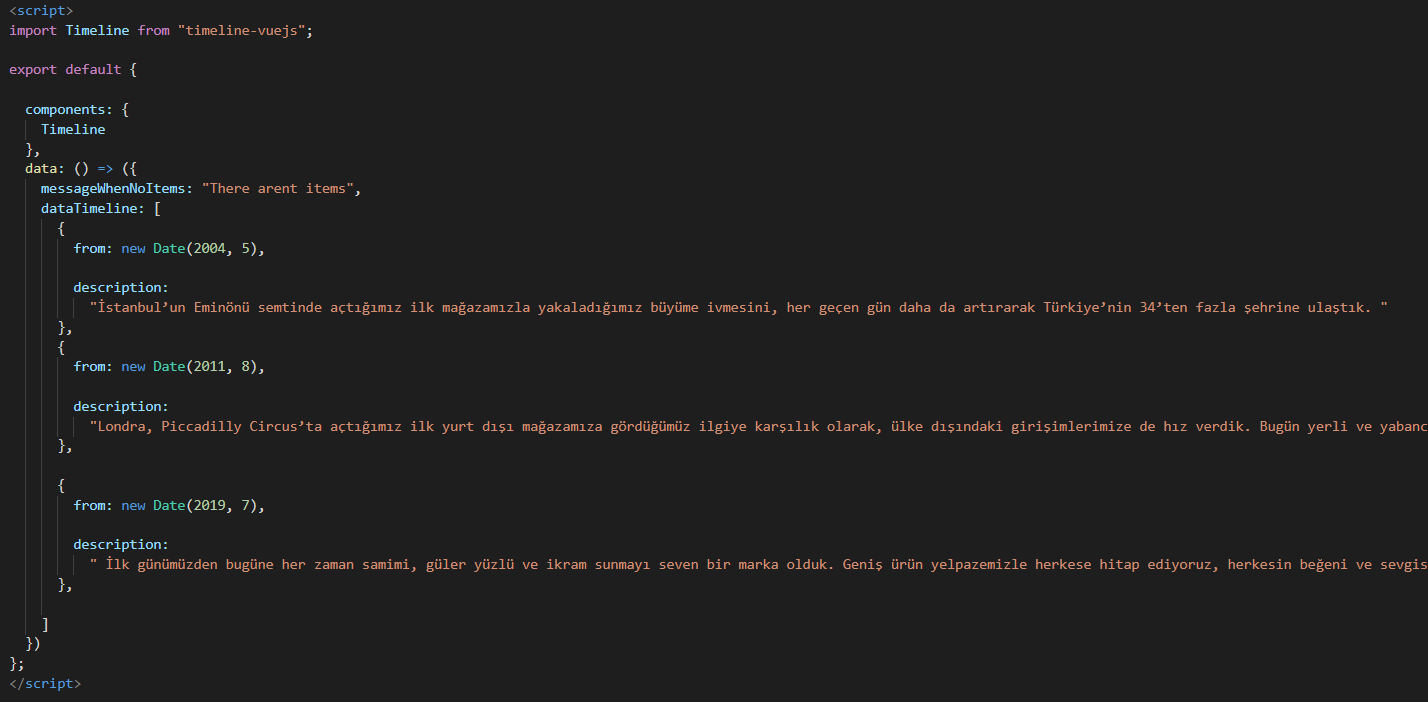
SpaceBetween , görseller arası boşluklar belirtilmiştir.

# **HAKKIMIZDA.VUE**

Timeline kullanılarak şirketin geçirdiği süreçler tanıtılmıştır.Böylece gerçekleşen olaylar tarihe göre sınıflandırılmıştır.



Order:asc ile eski zamandan günümüze doğru timeline sıralanmıştır. Desc yazarsak günümüzden eskiye doğru gidecektir.

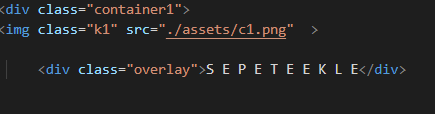


Timeline-vuejs kütüphanesi import edildi.

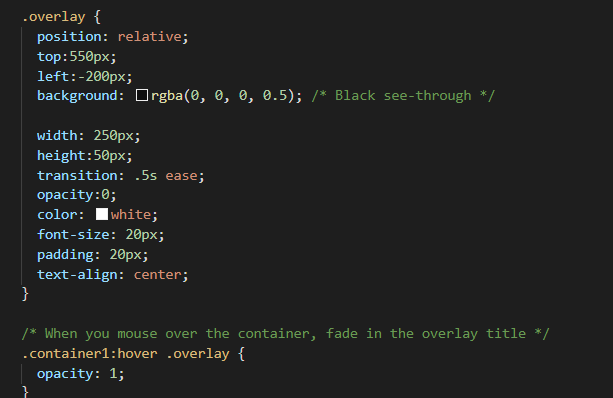
From: new Date(2004 , 5) :bu kısımda yıl ve ay belirtildi.

Description: Bu kısımda o yıla ait gerekli açıklamalar belirtildi.

# **KUTUCİKOLATA.VUE**



Ürünler bu şekilde sıralanmıştır.



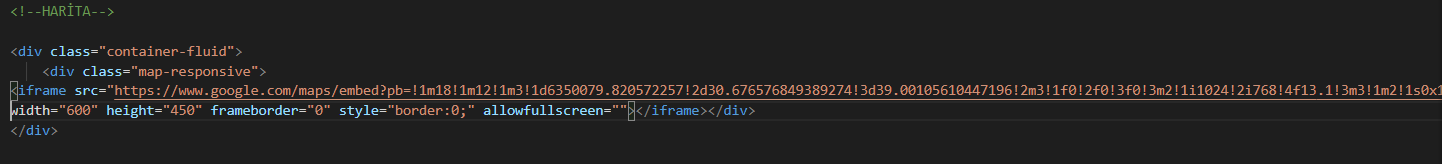
Transition: Sepete ekle butonuna gelince renk değişimi olmaktadır.Bunun geçiş hızı transition ile belirtilmiştir.

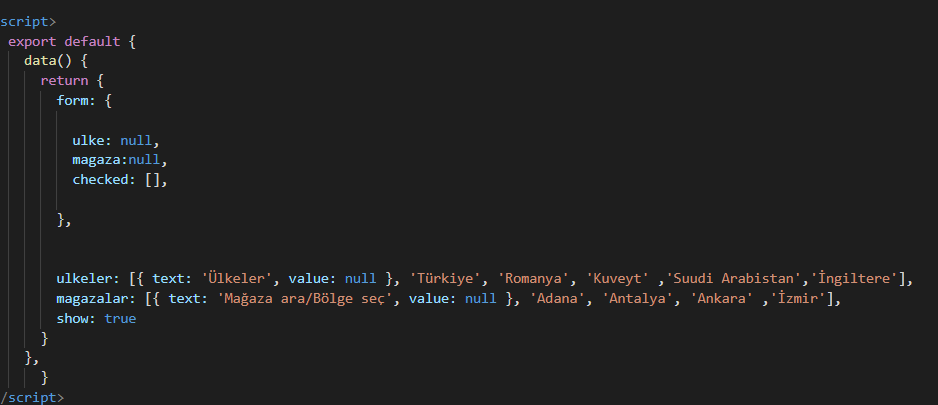
Text-align:center , yazının butonun ortasında durması sağlanmıştır.

Font-size: Yazının boyutu ayarlanmıştır.

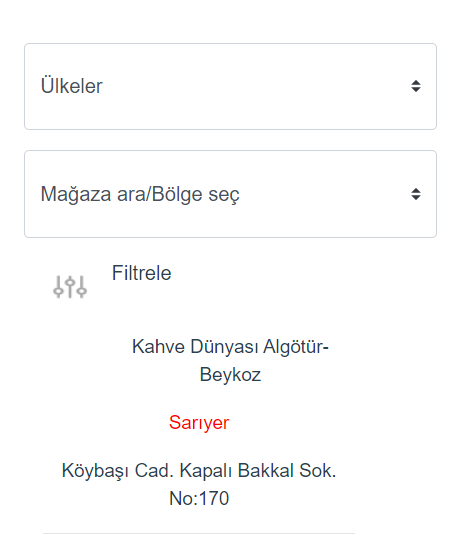
# **MAGAZALAR.VUE**

Mağazalar sayfasına iframe yapısı ile Türkiye haritasının aktif görseli eklenmiştir.



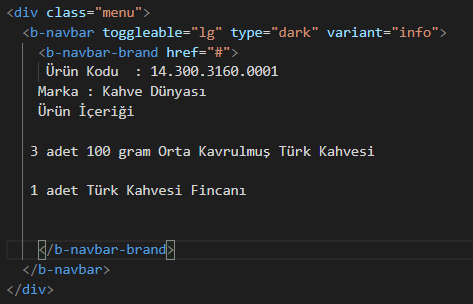
 

Liste oluşturuldu.Ülkeler , mağazalar formlarına gerekli değerler verildi.



# **ÜRÜNBİLGİ.VUE**

Sayfada aşağı doğru inildikçe , üst kısımda ürüne ait özet bilgiler gözükmektedir.Bu da sayfaya sabitlenmiştir.





b-card pills card vertical ile tab menü de tıklanılınca yatayda açılması belirtilmiştir.

Title: Solda bulunan seçeneklerin isimleri belirtilmiştir.

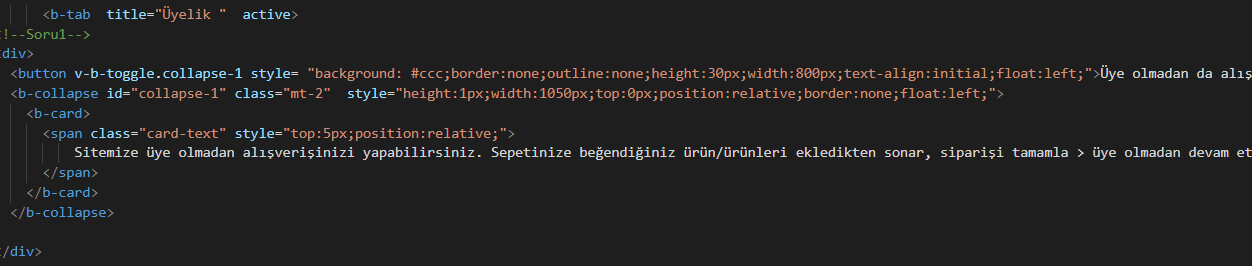
Br: Bu tag ile alt satıra geçme işlemi gerçekleşmektedir.

Strong:Bu tagin arasına yazılan yazı kalın bir şekilde olur.

Text-align:left ile yazı sola alınmıştır.

# **ÜYELİK.VUE**

Bu kısımda müşterilerin sorduğu sorular ve onlara verilen yanıtlar tanımlandı.



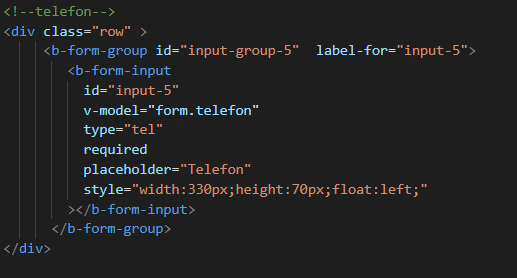
<button v-b-toggle.collapse-1> : Soru butonuna basıldığında collapse ile tanımlanan cevap içeriğini gösterir.Tekrar basıldığında içeriği gizler.

<b-collapse/>:İçeriğin gizlendiği kısım.

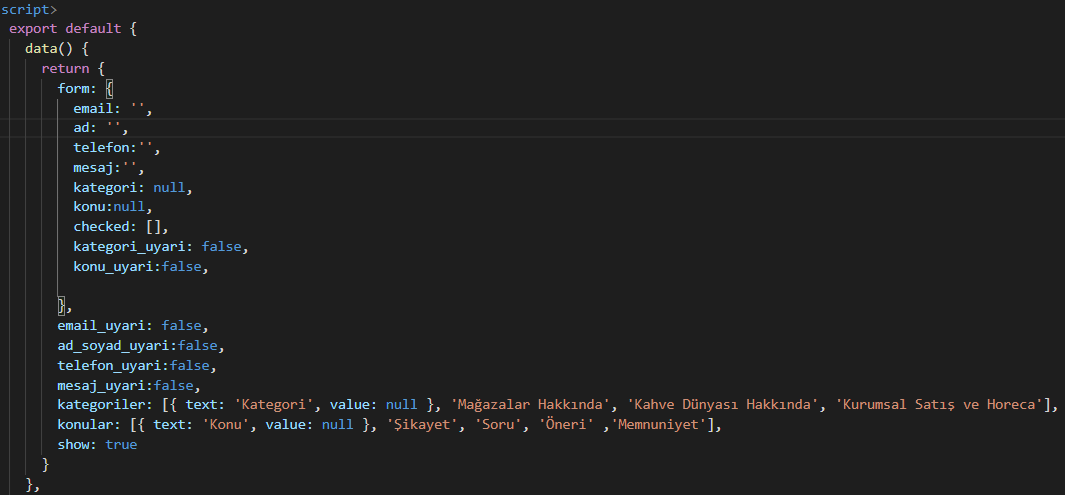
<b-card/> :Sayfa düzeni sağlandı.



Kategori formu’nun içerikleri liste ile tanımlandı.Kategori formuna tıklanıldığında seçme hakkı sağlanır.Kullanıcı istediği kategoriyi seçer.

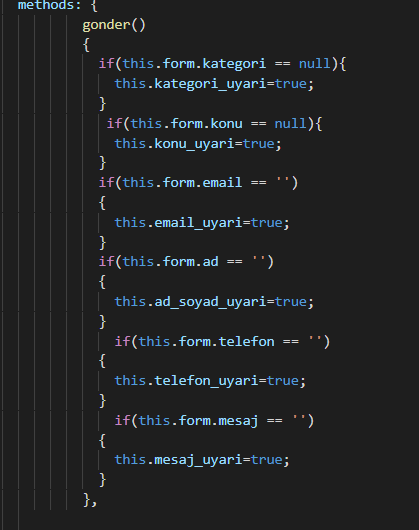


Kullanıcının telefon girdisi yapabilmesi sağlandı.v-model iledata içerisindeki form kısmındaki bilgilere ulaşıldı.Bilgiler doldurulmadığı takdirde gönder butonuna basıldığnda kullanıcıya uyarı verildi.



Kategoriler ve konular liste şeklinde tanımladı ve değerleri verildi.Boş bilgi girilirse kodumuz hata belirtiyor ve tekrar girilmesi isteniyor.

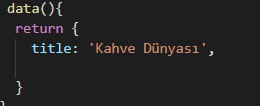
Kullanıcı girdisini kontrol edebilmek için metodlar tanımlandı.True ,false ile kontrol sağlandı.



# **BROWSER TAB DEĞİŞİMİ**

****

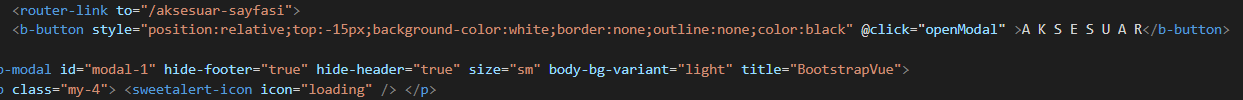
Çağrılan sayfamızın divine v-title ekliyoruz.

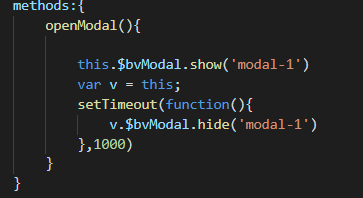


Divde belirttiğimiz title’ada Kahve Dünyası yazıyoruz böylece browser tab değişmiş oluyor.

Bu işlemi her açılan yeni sayfa için tekrarlıyoruz.

# **VUE-LOADİNG**

****

****

Öncelikle openModal isimli methodumuzu tanımlıyoruz.Bu metot sayesinde vue-loadingin süresi ayarlanmış oluyor biz 1 sn durmasını sağladık. Aksesuar butonuna tıklanınca openModal metodu çalışıyor ve sayfa yüklenirken giftimiz de ekranda gözükmüş oluyor.

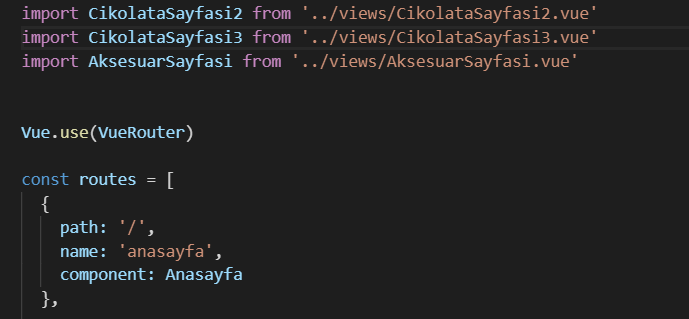
sweetalert-icon bizim gift tagimizdir. Burada icon ile belirtilen loading seçeneğini değiştirirsek giftimizde değişecektir.

**ROUTER KULLANIMI**

İlk olarak cmd ekranında vue create proje\_adi şeklinde proje oluşturulur.Önümüze gelen seçeneklerde manuel seçeneğini seçip ardından router’ı space ile seçip yeni projemizi oluşturmuş oluyoruz.



Tüm ilgili componentleri ekliyoruz.Buradaki componentleri açılmasını istediğimiz sayfalarda view bölümünde ekleyip,çağıracağız.



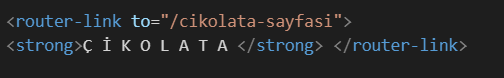
Open editors’da bulunan router sayfamızda yeni oluşturulacak sayfaları import ediyoruz ve link veriyoruz.Burada her view sayfasına link veriliyor ve tanımlamalar burada gerçekleşiyor.

’/’ Tek slash ile sayfanın anasayfa olduğunu belirtiyoruz.

Aynı şekilde view ekranında Anasayfa componentini oluşturup çağırmak istediğimiz componentleri burada belirtiyoruz.



Yukarıdaki görselde görüldüğü gibi Menu ekranında bulunan Cikolata seçeneğine tıklanınca açılan sayfada bulunması gereken componentleri ekliyoruz.

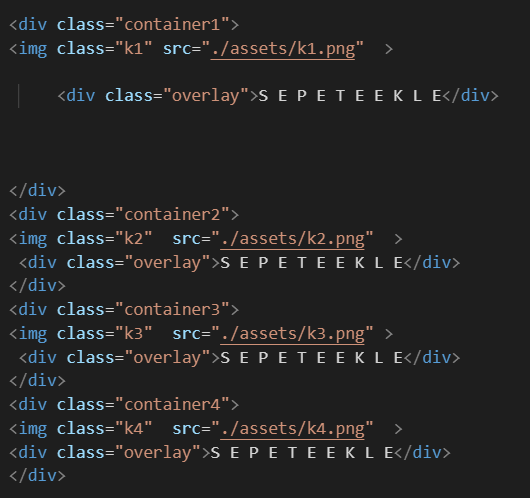


Bu kısımda router-link ile gidilen sayfanın linki veriliyor. Böylece Çikolata seçeneğine tıklanınca yeni sayfa gelmiş oluyor.Bizde view’de bulunan çikolata sayfasından componentleri çekmiş oluyoruz.

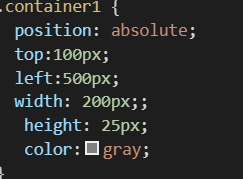
**KAHVE.VUE**

****

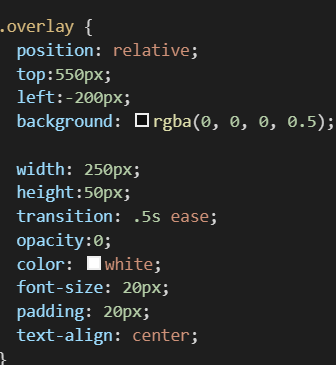
Bu sayfamızda Header,Menu ve Footer componentlerimiz sabit ve bunların arasında olan bölüm hazırlanmıştır. İlk olarak img tagi ile resim eklenmiştir.

****

Kahve sayfasında bulunan 96 ürün bu şekilde elenmiştir.

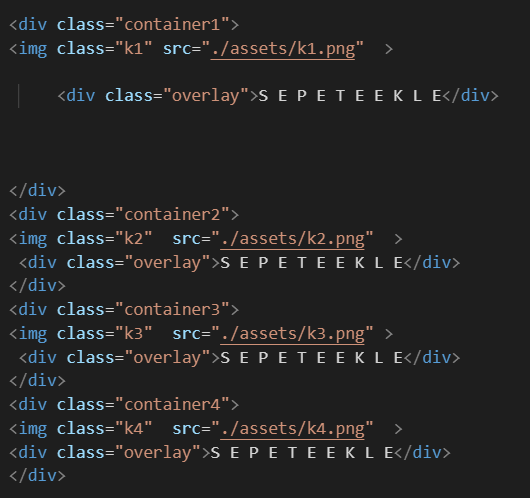
****

Container class ında img in üzerine gelinince çıkan sepete ekle butonunun özellikleri tanımlanmıştır. Color ile renk, position ile konum ayarı yapılmıştır.

****

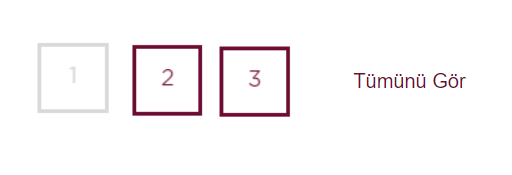
Transition .5s ile sepete ekle butonu geliş süresi belirtilir.

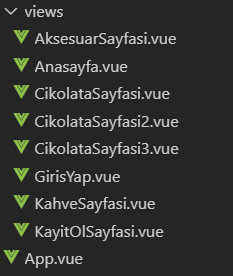
**CİKOLATA.VUE**

****

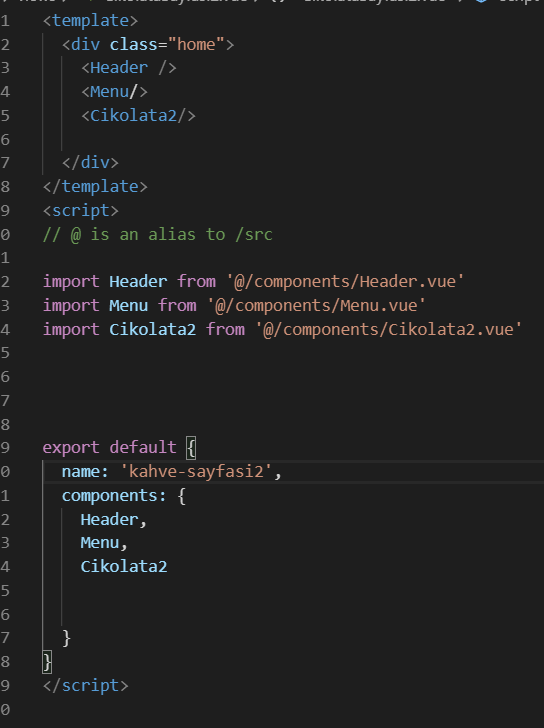
Kahve sayfasında olduğu gibi ürün eklenmiştir.







View’de CikolataSayfasi, CikolataSayfasi2 ve CikolataSayfasi3 adında componentler oluşturuldu.Burada yukarıdaki butonlara tıklanınca sayfaya yeni ürünlerin gelmesi sağlandı. Örneğin 2 butonuna tıklanınca CikolataSayfasi2 gelmektedir.

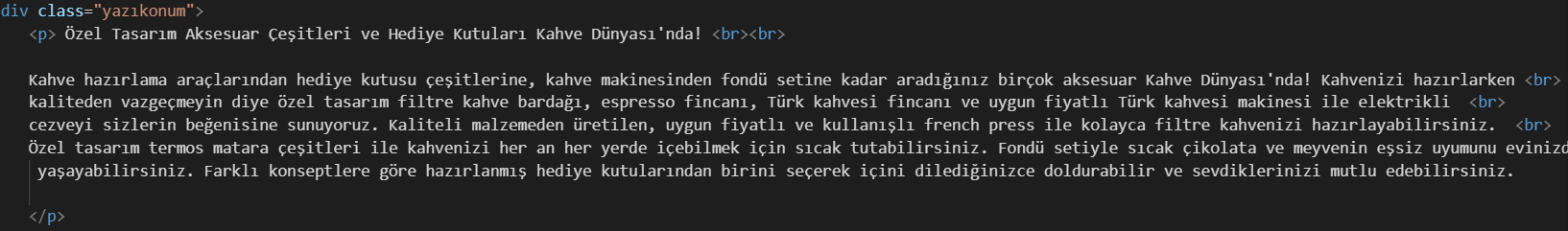


Yukarıdaki sayfada da çağırma işlemi gerçekleşmektedir.Açılmasını istediğimiz sayfaya hangi component orada çalışacak ise import edip,burada çağırırız.

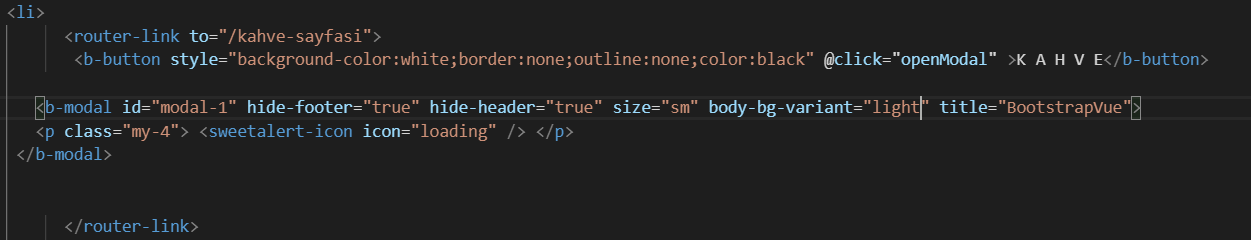
**AKSESUAR.VUE**

****

Ürünler sıralı olarak eklenmiştir.Diğer sayfalara ek olarak aşağıda belirtilen <p> tagine paragraf eklenmiş <br> tag i ilede alt satıra geçilmiştir.

****

**GIFT YAPIMI**



**Kaynak:** <https://www.npmjs.com/package/vue-sweetalert-icons>

Router-link ile gideceği sayfaya yönlendirme işlemi yapılır.

Kahve yazısına tıklanıldığında @click=”openModel” fonksiyonu çalıştırılıyor.

Hide-footer=true, hide-header=true ile modal da çıkan header ve footer bölümleri gizlendi.

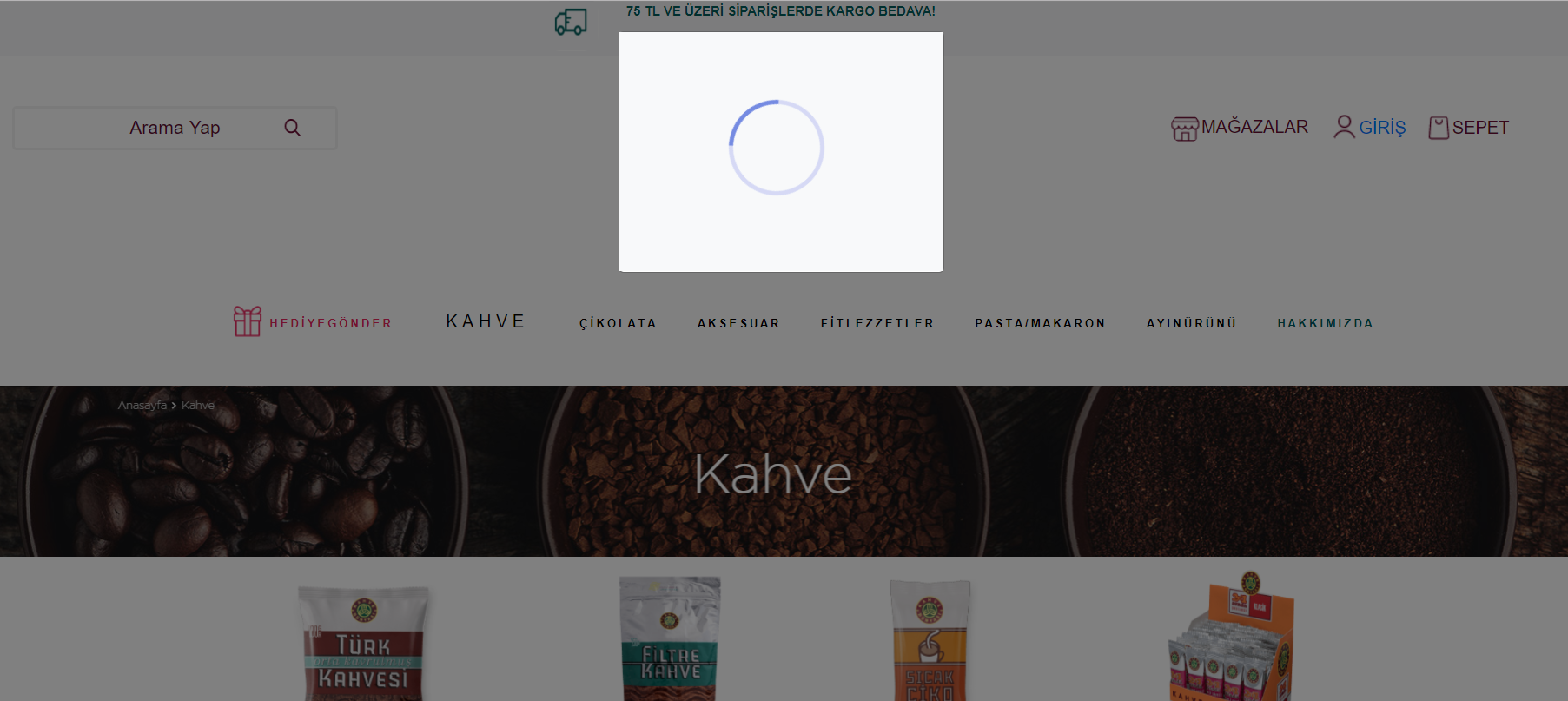
<sweetalert-icon> taginde animasyonumuz çağırılıyor ve icon=loading türü kullanılıyor.Loading yerine success gibi türlerde yazılabilir.

Son olarakta router-link tagi kapatılır.



OpenModel() fonksiyonu Menu.vue nin içerisinde tanımlanmıştır.Bu metot çalıştığında ilk olarak modal-1 isimli modal açılır daha sonra setTimeout fonksiyonu ile 1 saniye süre sonunda modal yani giftimiz gizlenmiştir.

Sonuç olarak aşağıdaki görüntü ortaya çıkmıştır:



**ARAMA BUTONU POPUP**



**Kaynak:** <https://bootstrap-vue.js.org/docs/components/modal>

V-model=modalShow adında bir veri gösterilmiştir.

Hazır alınan modal üzerinde değişiklikler yapılmıştır.

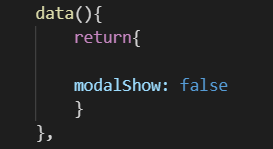
Örneğin;

Hide-footer=true ile footer gizlenmiştir,

Size=xl ile boyut ayarlanmıştır,

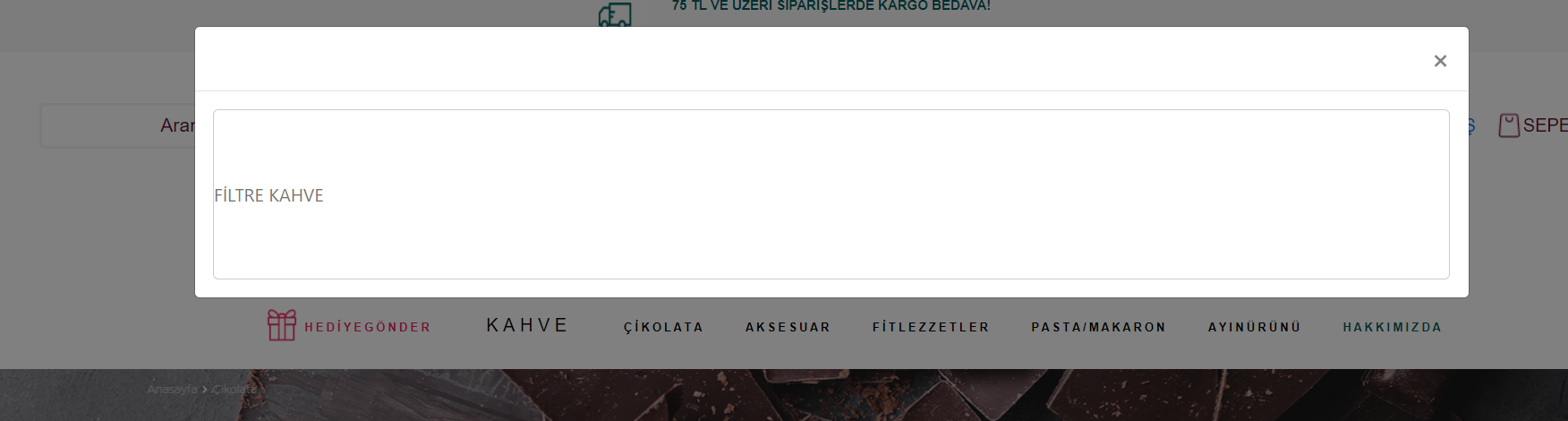
Outline ile border’ın etrafında hazır halde gelen mavi çizgi ortadan kaldırılmıştır.

Border:none border’ın sınırlarını ortadan kaldırmıştır.

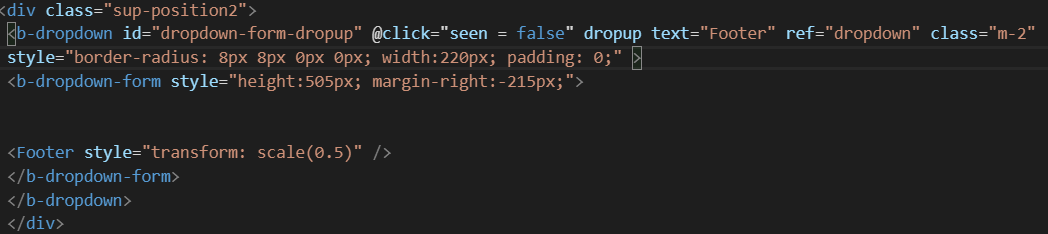


Modalshow:false özelliği ile modal, sayfa yüklenir yüklenmez gelmiyor.Butona tıklandığında arama butonu ekrana gelmiş oluyor.

Sonunda çıktımız aşağıdaki şekildeki gibi olur:



**FOOTER SABİTLEME**

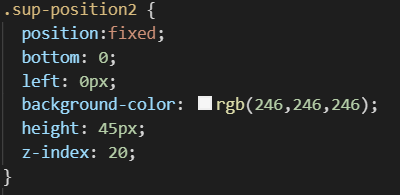


Kaynak: <https://bootstrap-vue.js.org/docs/components/dropdown>

Dropdown menü kullanılarak footer sabitlendi.

Border-radius ile border’ın etrafı yuvarlak yapıldı.

Transform:scale(0.5) ile boyut ayarlandı.



z-index ile görselin altta kalması önleniyor bu sayede footer anasayfada arka planda kalmıyor.

Position:fixed ile footer anasayfaya sabitlendi.

Sonunda çıktımız aşağıdaki şekildeki gibi olur:



# **PRE-LOADER YAPIMI**

 <meta http-equiv="refresh" content="4;URL=http://localhost:8081/anasayfa" />

        <center>

           <a href="https://imgflip.com/gif/3jvgy3"><img src="https://i.imgflip.com/3jvgy3.gif" title="made at imgflip.com"/></a>

        </center>

    </div>

View’de gift.vue adında bir component açılmıştır.Burada content’de loader yüklendikten sonra açılacak olan sayfa belirtilmiştir. src ‘de önceden oluşturmuş olduğumuz giftin adres bilgileri girilmiştir.Oluşturulan sayfanın path’i aşağıdaki şekilde oluşturulmuştur.

 {

    path: '/',

    name: 'anasayfa',

    component:Gift,

  },

# **TEKLİ VE ÇOKLU LİSTELEME**

Aksesuar başlığındaki ürünler aşağıdaki şekilde JSON’dan POST metodu ile çekilmektedir.Burada tek tek veriler alınıp,listelenmektedir.

<div  style="position:relative;top:110px;left:-550px;">

                     <!-- %100 -->

                       <div   v-for="(kahve,index) in Aksesuarlar" :key="index">

                            <!-- %25 -->

                            <a v-bind:href="kahve.url" >

                                <br> <br> <br> <br> <br>

                             <img  v-bind:src="kahve.image" />

                            </a>

                             <p style="position:relative;left:400px;top:-130px;"><strong> {{kahve.acıklama}} </strong> </p>

                             <p style="color:red;position:relative;left:600px;top:-165px;"><strong> {{kahve.fiyat}} </strong> </p>

                            <b-button style="position:relative;left:800px;top:-215px;">S E P E T E E K L E </b-button>

                        </div>

 <div class="row" style="position:relative;top:140px;">

                     <!-- %100 -->

                       <div class="col-3"  v-for="(kahve,index) in Aksesuarlar" :key="index">

                            <!-- %25 -->

                            <a v-bind:href="kahve.url" >

                                <br> <br> <br> <br> <br>

                             <img  v-bind:src="kahve.image" />

                            </a>

                             <p><strong> {{kahve.acıklama}} </strong> </p>

                             <p style="color:red;"><strong> {{kahve.fiyat}} </strong> </p>

Yukarıdaki ifadede ise ek olarak sadece satır ve sütun bilgisi girilerek ürünler 4’lü olarak listelenir.

# **SERVER OLUŞTURMA**

## **İNDEX.JS**

Server oluştururken öncelikle kullanacağımız modülleri ekliyoruz.

var express = require('express')

Bunlardan ilki Express modülüdür.Express, Node.js tabanlı bir web sunucu çatısıdır.

var cors = require("cors")

Cors bir web sayfası üzerindeki bazı kaynakların, kaynağın sunulduğu alan adının dışındaki bir alan adından istenebilmesine izin veren bir mekanizmadır.

var bodyParser = require("body-parser")

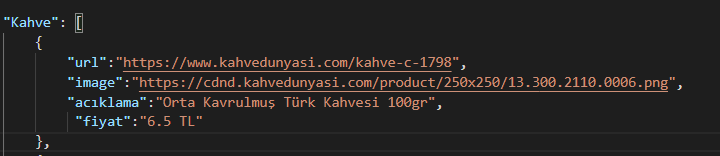
bodyParser ise gönderilen post datasını obje olarak yakalamamızı sağlayan bir modüldür.Ardından aşağıda görüldüğü gibi hangi uzantılar için hangi route objesinin kullanılacağı belirtilir.

app.use("/turk", turk)

## **DATA.JSON**

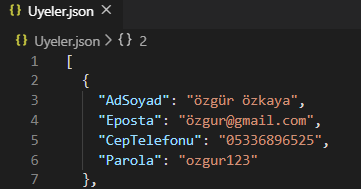
JSON; bütün programlama dilleri arasında, yapılandırılmış veri değişimini kolaylaştıran bir metin biçimidir.JSON’da Kahveler adı altında bir liste oluşturuldu ve kahvenin url,image,açıklama ve fiyat bilgileri girildi.

url’de sayfanın kendi linki image’de ise görselin adresi tutulmuştur.



## **UYELER.JSON**

Kullanıcılara ait veriler burada tutulur.GET ile buradaki veriler okunur POST ile de veriler buraya eklenir.



## **HOME.JS**

İki tane fonksiyonumuz vardır. Home.get ve home.post fonksiyonları iki parametre alır request(req) ve response(res). Request yolladığımız istektir ve response bize isteğimiz doğrultusunda dönen cevaptır.

Home.js sayfasında GET ve POST metotlarımız bulunur. GET metodu verileri almamızı sağlar.

ReadFileSync ile dosyamızı okuruz.

Json.parse ile dosyadan okunan veriler dönüştürülür.

Res.send ile de veriler gönderilir.



## **UYE.JS**

Burada POST metodu ile web sayfasından gönderilen veriler alınır. İlk olarak liste oluşturulur.

ReadFileSync ile dosya okunur.Arr.push ile üyelere, yeni eklenen üyeler eklenir.

writeFileSync ile veriler yazılır.

