

Front-End Bootcamp Final Project

**Hami Berkay Aktaş**

# İçindekiler

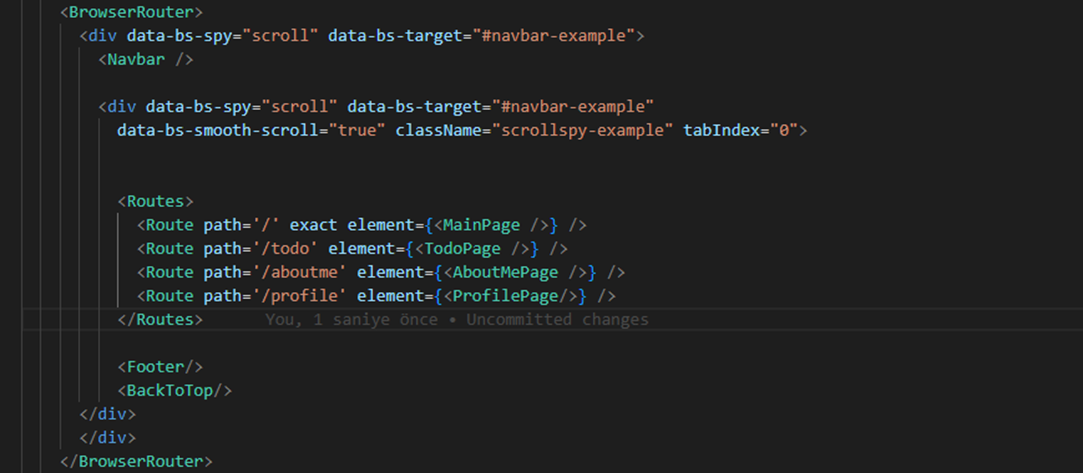
# React Kütüphaneleri

1. lottiefiles/react-lottie-player 3.4.7
2. mui/material 5.9.0
3. reduxjs/toolkit 1.8.3
4. axios 0.27.2
5. boostrap 2.0.0
6. formik 2.2.9
7. react 18.2.0
8. react-animated-numbers 0.13.0
9. react-circular-progressbar 2.1.0
10. react-hot-toast 2.3.0
11. react-redux 8.0.2
12. react-router-dom 6.3.0
13. react-wow 1.0.0
14. wowjs 1.1.3
15. yup 0.32.11

# 1 Componentlerin React Yapısına Uygun Şekilde Jsx’e Çevrilmesi

İlk önce clean kodumuzu html to jsx çevirici ile reacte uygun hale getirdim ardından sub componentler olaralar Components klasörü alttında topladım. Ana yapıyı inşa ettikten sopnra index htmlye boostrap cdn inin ekledim enson olarakda react router dom kurarak site iskeletini ayağa kaldırmış oldum

# 2. React Router Dom Kurulumu ve Routeları tanımlama



Projemizde bir giriş kısmı var ana sayfa bir adet todo sayfamız bir adet profile sayfası login ve register kısmından ötürü ve son olarak da about me kısmımız bulunuyor

Bu route larımızı ve browser router kısmımızı ayarladıktan sonra bu sayfamızı oluşturruruz sonra burada routes komponenti içinde renderlarız

# 3 Her sayfada olan komponentler

## 3.a Navbar

## 3.b Footer

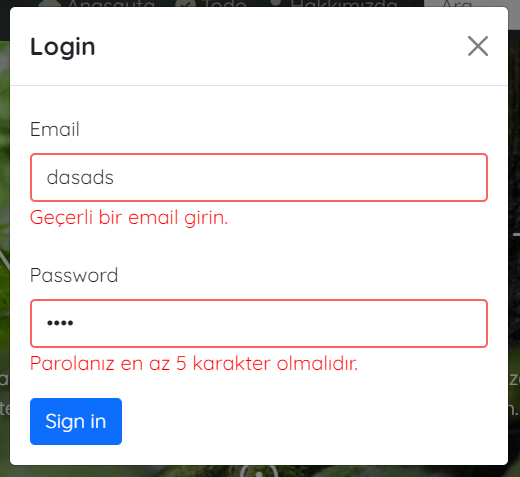
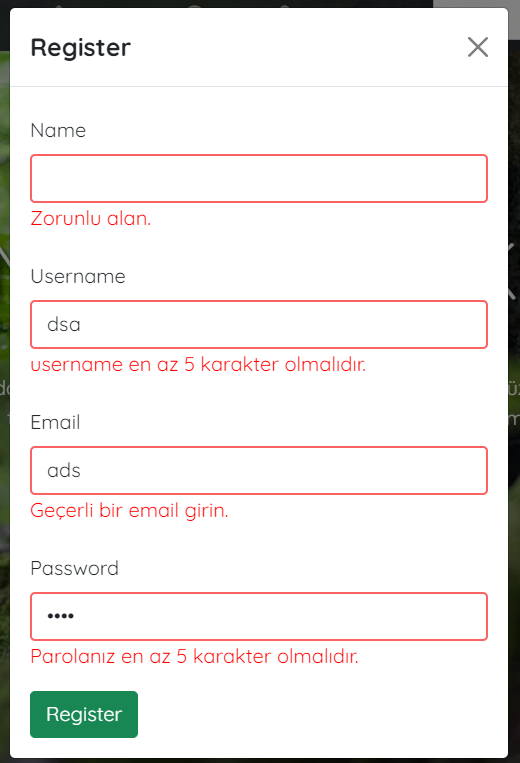
## 3.c BacktoTop

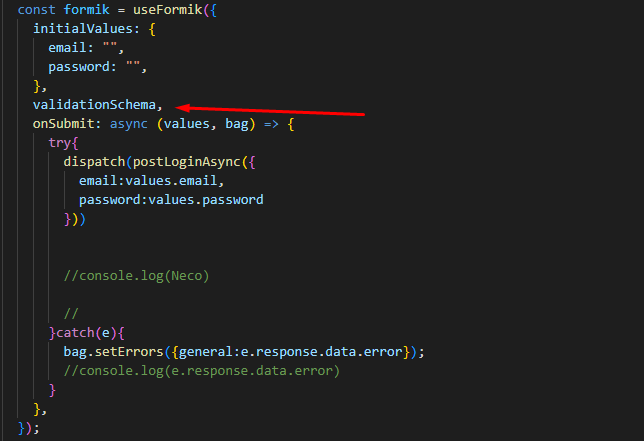
## 3.a Navbar

Bu komponentimizde ilk olarak rect router dom yapılarından link yapısını kullanacağız

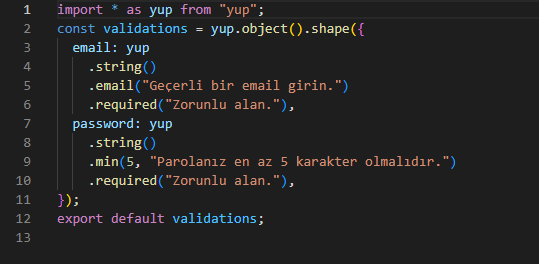
Ardından arama kısmında outo complete için material ui auto completre componentini kullanacağız

İcon kısımlarında font awesome kullandık ve modal yapılarında önyüz validasyonu yapmak için yup validatör ve formik kullandım



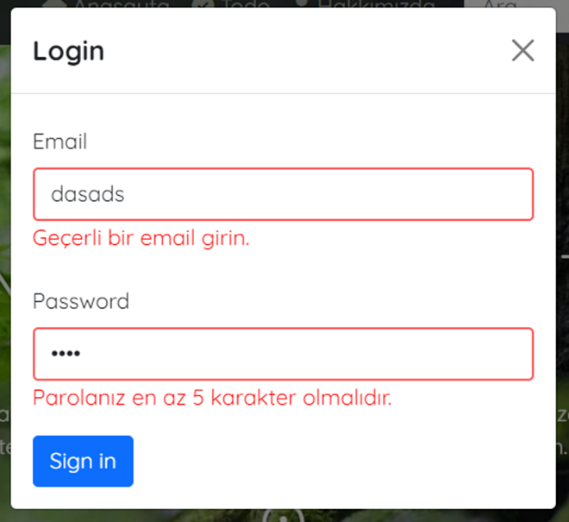


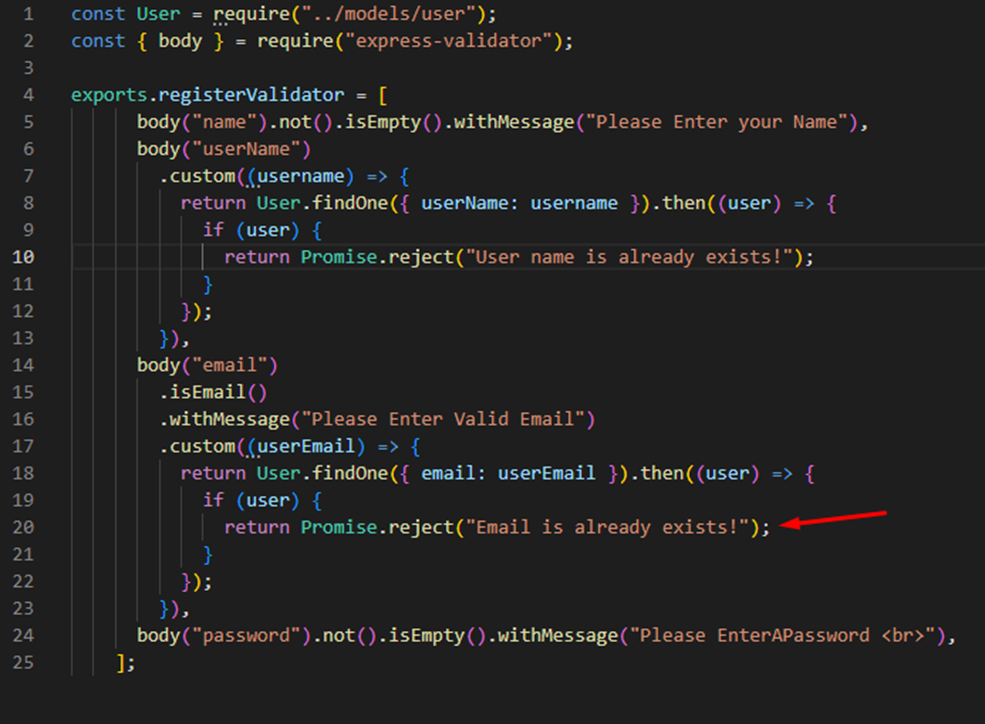
Formik içerisinde yup olarak validation shema tanımlanır (aşağıda göstermekteyim)



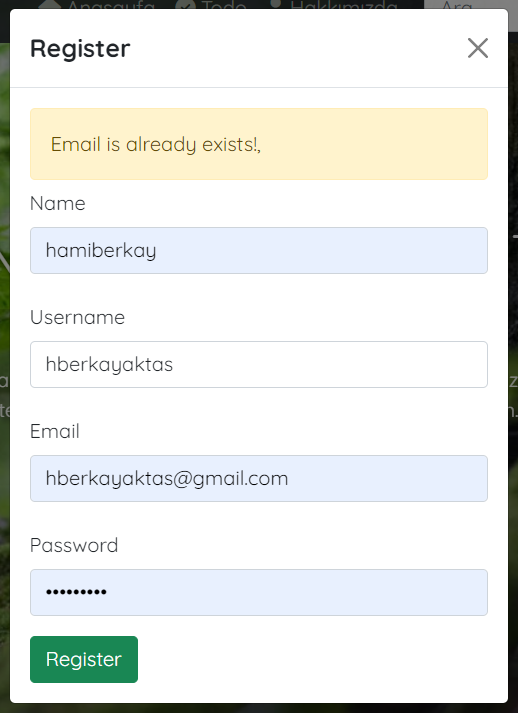
Yup dosyamızda neler istiyoruz bunları yazarız.

Burada yupdan gelen hatayı formik error bag e atarım ve ardından eğer kullanıcı input field ‘a dokunup boş ayrıldıysa hatayı bastırırım ve input field’ın çevresinin kırmızı yaparım.



Önyüz ve filtrelemelerini yaptıktan sonra arka planda validation işlemi için Express-validator ile validation yapıyoruz burada custom bir Validation error’u ekleyebiliyoruz. 

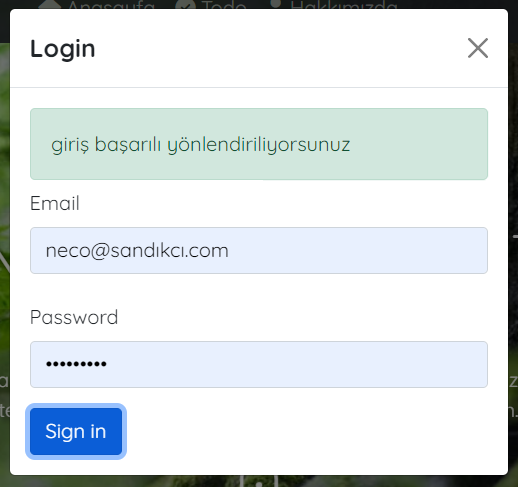
Hatanın geri çıktısı



Veritabanında aradıktan sonra uniq olsun diye hata geri döner.

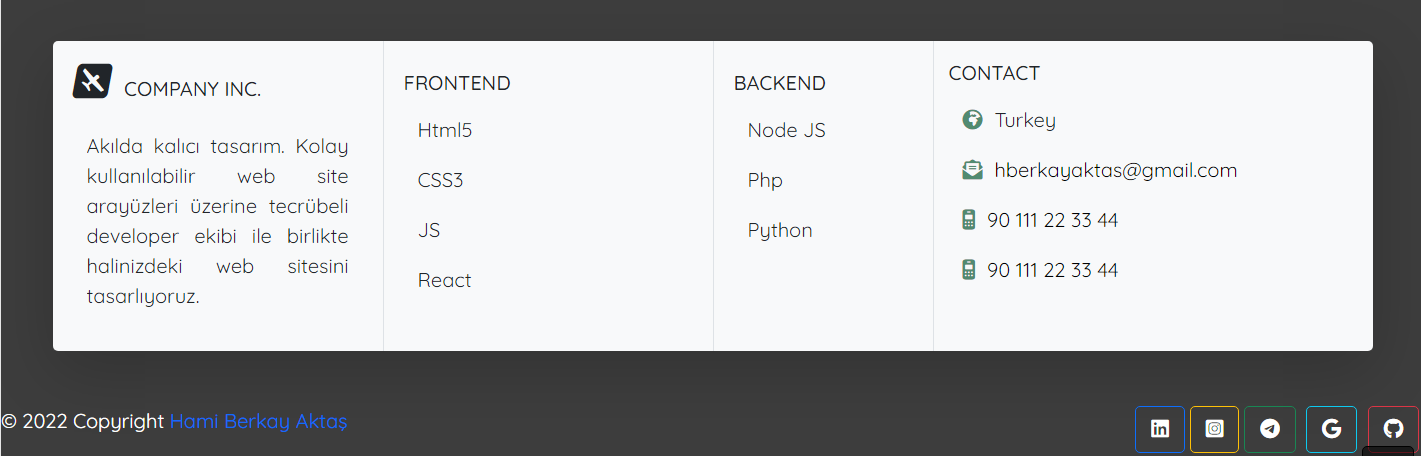
Eğer başarılı bir giriş yapıldı ise react router dom elemanlarından

Use navigate ile sayfa yönlendirmesi yapılır.



## 3.b Footer

Footer kısmında özet bir bilgi ekleyip görsel düzenlemeler yaptım



## 3.c BacktoTop





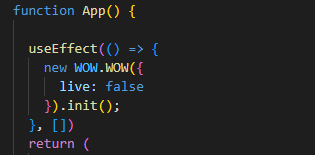
Bu komponent içerisinde ilk önce sayfa konumunu öğreniyoruz ardından eğer konumumuz pixel cinsinden 100 pixel’in altındaysa sayfanın başlarındayız demektir baack to topo elemanımızxı saklıyoruz eğer 100 pixel den büyük ise komponentimizi tekrardan gösteriyoruz.

# 4. Header sayfası ve ilk wow js entegrasyonu

Bu componentden itibaren wow js ve animasyon css kütüphanesine ihtiyaç duyarız burada ilk olarak direkt olarak npm install olarak kurarız

Npm install wowjs

Ardından use efffect ile did mount anında wow js yi initialize ederiz.

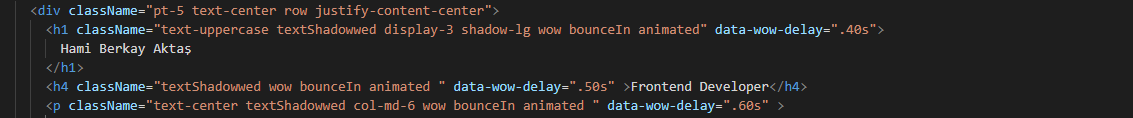


Ardından public klasörü altındaki index html ye animate .css eklenir.

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/animate.css@3.7.2/animate.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.0.0/animate.compat.css"/>

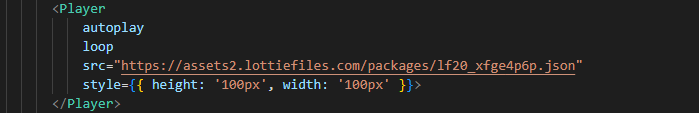
Ardından class parametrelerine eklenir



Ayrıca bu sayfada soll down animasyonunu eklemek için

Npm i react-lottie-player eklenir

Ardınan component içerisine şu şekilde bir şey eklenir





# 5.Ability Componenti

npm i react-circular-progressbar diyerek dairesel progress bar oluşturmak için kütüphane kurarız

kod yapımız aşağıdaki gibi oldu

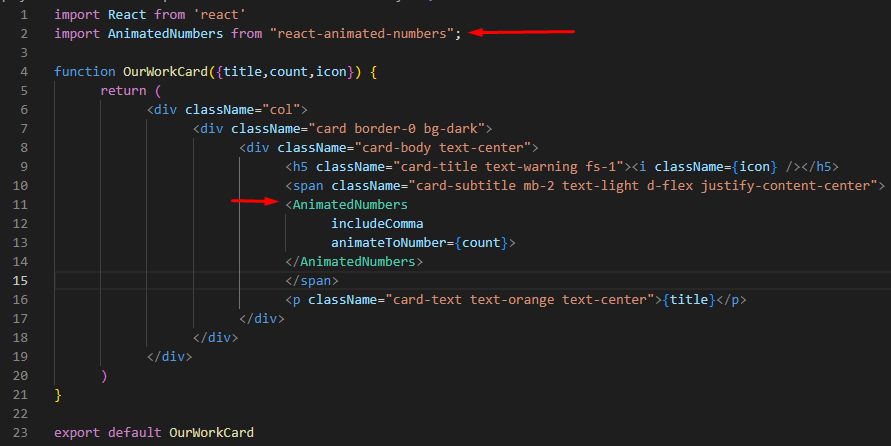




# 6.React animated number

Bu komponentde bu güne kadar yapıtığım çalışmalardan bilgileri animasyonlu rakamlar ile birlikte renderleriz kod yapısı ve görünüm aşağıdaki gibi olur

npm i react-animated-numbers





# 7. Todo Sayfam

Buradaki todo’ları React Redux state yönetim aracı ile yönetiriz.

Kodların içindeki todo slice içeriğinde daha detaylı şekilde todoSlice.js dosyasından erişe biliriz.

