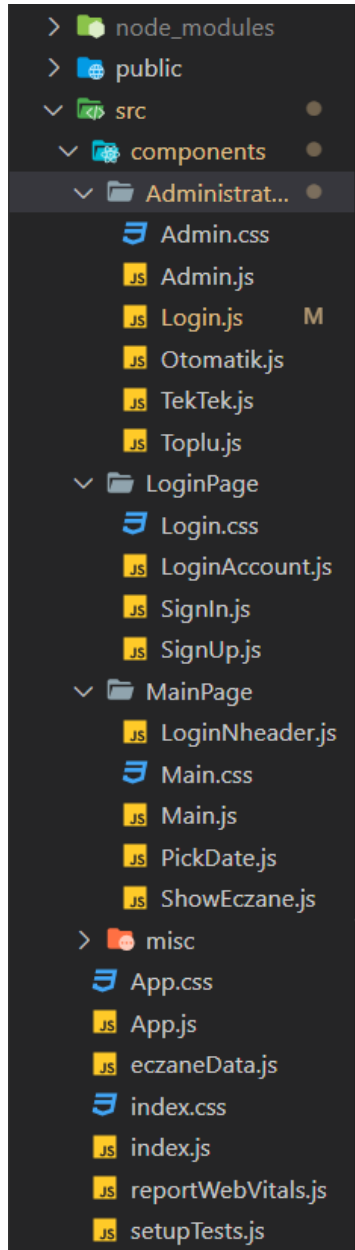


Seri Eczane Bul

Ahmet Fehim Örnek

1191602071



Projenin Yapısı

Ana State'ler

```
{"Merkez1":"-", "Merkez2":"-", "Meriç":"-", "Havsa":"-", "Enez":"-", "Keşan":"-", "Lalapaşa":"-", "Süloğlu":"-", "İpsala":"-", "Uzunköprü":"-"}
```

```
const [eczaneler, setEczaneler] = useState(EczaneList)
const [users, setUsers] = useState({admin : ["admin", "-", "-"], fehim: ["1234", "05454239292", "fehimornek@gmail.com"]})
```

Eczaneler sadece Admin panelinden ayarlanabilir.
Kullanıcılar ise login ekranında doğrulama yapma amaçlı tutulur.

Yardımcı State'ler

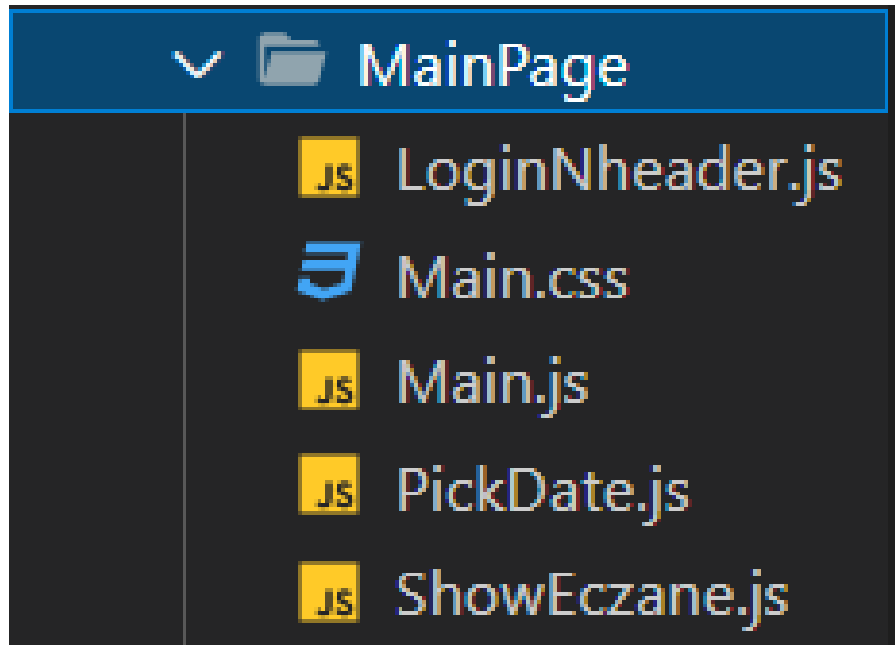
```
const [showDate, setShowDate] = useState(0)
const [newEczane, setNewEczane] = useState(0)
```

showDate hangi tarihe ait nöbetçi eczanenin seçileceğini gösterir.
newEczane yeni bir eczane gösterileceği zaman set edilir.

Router Mantığı

```
<Router>
  <div className="App">
    <Routes>
      <Route path="/" element={<Main eczaneler={eczaneler}/>}/>
      <Route path="/admin" element={<Admin eczaneler={eczaneler} setEczaneler={setEczaneler}/>}/>
      <Route path="/login" element={<LoginAccount users={users} setUsers={setUsers}/>}/>
      <Route path="*" element={<ErrorPage/>}/>
    </Routes>
  </div>
</Router>
```

1. Main component'inin url'si => /
2. Admin componentinin urlsi => /admin
3. Login componentinin urlsi => /login
4. Varolmayan bir sayfa gelirse ErrorPage adlı sayfa görüntülenir.



MainPage Component

Seri Eczane Bul

login

Gün veya tarih seçin

Bugünkü eczane

Yarınki eczane

06/09/2022



setShowDate()
setNewEczane()

Eczaneler

Şehir	Eczane
Merkez	Akın
Merkez	Altierler
Meriç	Öden
Havsa	Derman
Enez	Sağlık
Keşan	Akar
Lalapaşa	Sevil
Süloğlu	İnci
İpsala	Çınar
Uzunköprü	Arzu

Önemli Noktaları

Gün veva tarih seçin

June 2022 ▾ ↑ ↓

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Clear Today

06/09/2022



Gün veva tarih seçin

July 2022 ▾ ↑ ↓

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Clear Today

07/07/2022



Tarih Seçme Kodu

```
const minMaxDate = () => {  
  const fullDate = new Date()  
  let day = fullDate.getDay()  
  let month = fullDate.getMonth() + 1  
  let year = fullDate.getFullYear()  
  let date = fullDate.getDate()  
  var lengthOfMonth = new Date(year, month, 0).getDate();  
  
  let nextMonthPickableDays = 30 - (lengthOfMonth - date + 1)  
  let maxDate = lengthOfMonth  
  let maxYear = year  
  let maxMonth = month  
  
  // if it's not the beginning of the month (which is very unlikely)  
  if (nextMonthPickableDays > 0) {  
    maxDate = nextMonthPickableDays  
    maxMonth = month + 1  
    if (maxMonth === 13) {  
      maxMonth = 1  
      maxYear = year + 1  
    }  
  }  
  
  if (date < 10) date = "0" + date  
  
  if (month < 10) month = "0" + month  
  
  if (maxMonth < 10) maxMonth = "0" + maxMonth  
  
  if (maxDate < 10) maxDate = "0" + maxDate  
  
  const minimumDate = year + "-" + month + "-" + date  
  const maximumDate = maxYear + "-" + maxMonth + "-" + maxDate  
  return [minimumDate, maximumDate, lengthOfMonth]  
}  
  
export default minMaxDate
```


✓  LoginPage

 Login.css

 LoginAccount.js

 SignIn.js

 SignUp.js

LoginPage Component

main
page

Kullanıcı Adı

Şifre

giriş yap

Kullanıcı adı

Şifre

Telefon

Email

kayıt ol

```

const checkValid = (e) => {
  e.preventDefault()
  const username = document.getElementById("username").value
  const password = document.getElementById("password").value
  let match = 0
  Object.entries(users).forEach(([key, value]) => {
    if (key === username && value[0] === password) {
      navigator("/admin")
      match = 1
    }
  })
  if (match === 0) {
    document.getElementById("mesaj_hata_1").innerHTML = "şifre veya kullanıcı adı hatalı!"
  }
  setTimeout(function(){
    document.getElementById("mesaj_hata_1").innerHTML="";
  },1500);
}

```

Giriş yapma esnasında bilgilerin doğruluğunu test eden kod parçası.

```

else {
  Object.entries(users).forEach(([key, value]) => {
    if (key === username) {
      document.getElementById("mesaj_hata_2").innerHTML = "Seçilmemiş bir kullanıcı adı seçiniz!"
      error = 1
    }
    if (value[2] === email) {
      document.getElementById("mesaj_hata_2").innerHTML = "Bu email zaten kullanımda!"
      error = 1
    }
  })
}

```

Kayıt olma esnasında seçilen kullanıcı adı ve email'in tekrar etmesini engelleyen kod parçası.

Hata Kontrolleri

Kullanıcı Adı

asd

Şifre

...

giriş yap

şifre veya kullanıcı adı hatalı!

Kullanıcı adı

asddd

Şifre

...

Telefon

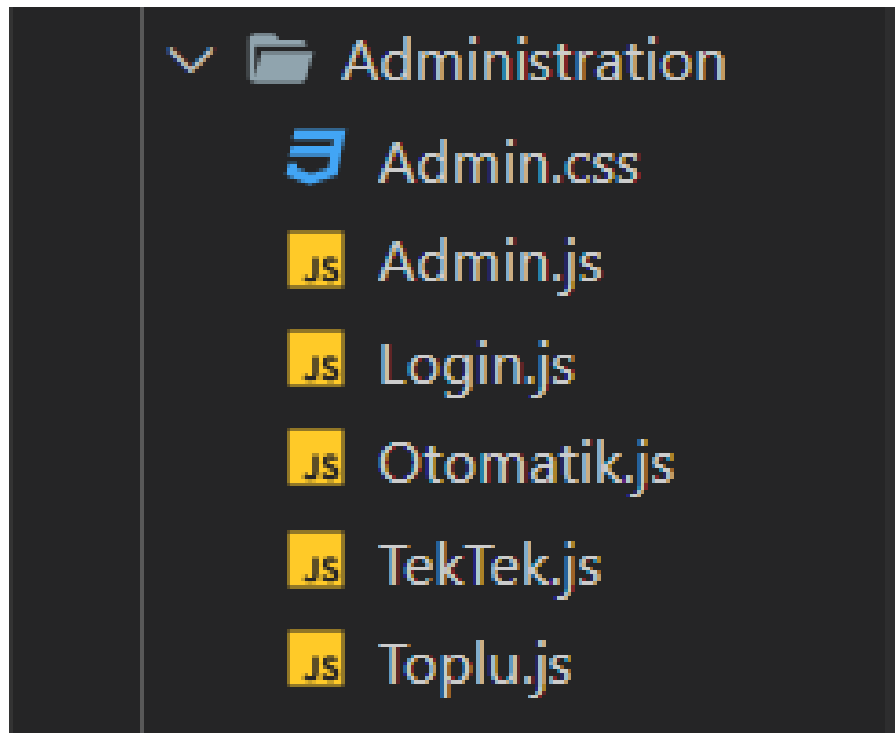
asaaa

Email

fehimornek@gmail.com

kayıt ol

Bu email zaten kullanımda!



Administration Component

main page

Admin Sayfası

Tek Tek Ata

Merkez1



mm/dd/2022



eczane ata

Toplu Ata

mm/dd/2022



Merkez1

Merkez2

Meriç

Havsa

Enez

Keşan

Lalapaşa

Süloğlu

İpsala

Uzunköprü

eczane ata

Otomatik Ata

1 aylık eczane ata

Admin Sayfası Genel Üç Component

```
<Login></Login>  
<TekTek eczaneler={eczaneler} setEczaneler={setEczaneler} dates={dates}></TekTek>  
<Toplu eczaneler={eczaneler} setEczaneler={setEczaneler} dates={dates}></Toplu>  
<Otomatik eczaneler={eczaneler} setEczaneler={setEczaneler} dates={dates}></Otomatik>
```

Tek tek atama

```
const eczaneAtaTek = (e) =>{
  e.preventDefault()
  let seciliSehir = document.getElementById("sehir_sec_tek").value;
  let seciliEczane = document.getElementById("eczane_sec_tek").value;
  let index = indexCalculator("tarih_sec_tek", minDate, lenMonth)

  if (!Number.isNaN(index) && seciliEczane !== ""){
    const updatedEczane = eczaneler
    updatedEczane[index][seciliSehir] = seciliEczane
    setEczaneler(updatedEczane)
    document.getElementById("errorMessage").style.visibility = "visible";
    document.getElementById("errorMessage").innerHTML="Degisiklikler kaydedildi!";
  }

  else{
    document.getElementById("errorMessage").style.visibility = "visible";
    document.getElementById("errorMessage").innerHTML="Gerekli alanlar doldurunuz!";
  }
  setTimeout(function(){
    document.getElementById("errorMessage").style.visibility = "hidden";
  },1000);
}
```


Toplu Atama

```
const eczaneAtaToplu = (e) =>{
  e.preventDefault()
  let index = indexCalculator("tarih_sec_toplu", minDate, lenMonth)
  let Merkez1 = document.getElementById("Merkez1_inp").value
  let Merkez2 = document.getElementById("Merkez2_inp").value
  let Meric = document.getElementById("Meric_inp").value
  let Havsa = document.getElementById("Havsa_inp").value
  let Enez = document.getElementById("Enez_inp").value
  let Kesan = document.getElementById("Kesan_inp").value
  let Lalapasa = document.getElementById("Lalapasa_inp").value
  let Suloglu = document.getElementById("Suloglu_inp").value
  let Ipsala = document.getElementById("Ipsala_inp").value
  let Uzunkopru = document.getElementById("Uzunkopru_inp").value
  let updated_eczaneler = eczaneler

  if (!Number.isNaN(index) && Merkez1 !== "" && Merkez2 !== "" && Meric !== "" && Havsa !== "" && Enez !== "" && Kesan !== "" && Lalapasa !== "" &&
    updated_eczaneler[index]["Merkez1"] = Merkez1
    updated_eczaneler[index]["Merkez2"] = Merkez2
    updated_eczaneler[index]["Meric"] = Meric
    updated_eczaneler[index]["Havsa"] = Havsa
    updated_eczaneler[index]["Enez"] = Enez
    updated_eczaneler[index]["Kesan"] = Kesan
    updated_eczaneler[index]["Lalapasa"] = Lalapasa
    updated_eczaneler[index]["Süloğlu"] = Suloglu
    updated_eczaneler[index]["İpsala"] = Ipsala
    updated_eczaneler[index]["Uzunköprü"] = Uzunkopru

    document.getElementById("errorMessage").style.visibility = "visible";
    document.getElementById("errorMessage").innerHTML="Degisiklikler kaydedildi!";
  }
  else {
    document.getElementById("errorMessage").style.visibility = "visible";
    document.getElementById("errorMessage").innerHTML="Gerekli alanları doldurunuz!";
  }
  setTimeout(function(){
    document.getElementById("errorMessage").style.visibility = "hidden";
  },1000);
  setEczaneler(updated_eczaneler)
}
```

Otomatik Atama

```
let secilebilirEczaneler = {merkez:["Akın", "Altın", "Aytaç", "Baca", "Baraj", "Begüm", "Berkay", "Çakbil", "Can", "Çebe", "Cent", "Çiftçi", "Deniz", "Denizim", "De  
meric:["Öden", "Yeni Meriç"],  
havsa:["Derman", "Edirne", "Gün", "Havsa", "Yavuz"],  
enez:["Sağlık", "Hayat", "Saygın"],  
kesan:["Akar", "Alkan", "Banguoğlu", "Bilge Nur", "Birgül", "Deniz", "Ebru", "Erkan", "Gökhan", "Gölemen", "Güven", "Hayat", "Kanak", "Kural  
lalapasa:["Sevil", "Beyhan"],  
suloglu:["İnci", "Süloğlu"],  
ipsala:["Çınar", "Coşkun", "Devranoğlu", "Uysal"],  
uzunkopru:["Arzu", "Aylin", "Aytin", "Başak", "Bizim", "Çağla", "Deva", "Gencan", "Güneş", "Halit Orhon", "Hayat", "Manga", "Meriç", "Müge"]
```

```
let indexes = {merkez:0, meric:0, havsa:0, enez:0, kesan:0, lalapasa:0, suloglu:0, ipsala:0, uzunkopru:0}
```

Otomatik Atama

```
const birAyAta = () => {
  let updated_eczaneler = eczaneler
  for (let i = 0; i < 30; i++){
    updated_eczaneler[i].Merkez1 = secilebilirEczaneler.merkez[indexes.merkez]
    updated_eczaneler[i].Merkez2 = secilebilirEczaneler.merkez[indexes.merkez + 1]
    indexes.merkez += 2
    updated_eczaneler[i].Meriç = secilebilirEczaneler.meric[indexes.meric]
    indexes.meric += 1
    updated_eczaneler[i].Havsa = secilebilirEczaneler.havsa[indexes.havsa]
    indexes.havsa += 1
    updated_eczaneler[i].Enez = secilebilirEczaneler.enez[indexes.enez]
    indexes.enez += 1
    updated_eczaneler[i].Keşan = secilebilirEczaneler.kesan[indexes.kesan]
    indexes.kesan += 1
    updated_eczaneler[i].Lalapasa = secilebilirEczaneler.lalapasa[indexes.lalapasa]
    indexes.lalapasa += 1
    updated_eczaneler[i].Süloğlu = secilebilirEczaneler.suloglu[indexes.suloglu]
    indexes.suloglu += 1
    updated_eczaneler[i].İpsala = secilebilirEczaneler.ipsala[indexes.ipsala]
    indexes.ipsala += 1
    updated_eczaneler[i].Uzunköprü = secilebilirEczaneler.uzunkopru[indexes.uzunkopru]
    indexes.uzunkopru += 1

    if (indexes.merkez >= secilebilirEczaneler.merkez.length) indexes.merkez = 0
    if (indexes.meric >= secilebilirEczaneler.meric.length) indexes.meric = 0
    if (indexes.havsa >= secilebilirEczaneler.havsa.length) indexes.havsa = 0
    if (indexes.enez >= secilebilirEczaneler.enez.length) indexes.enez = 0
    if (indexes.kesan >= secilebilirEczaneler.kesan.length) indexes.kesan = 0
    if (indexes.lalapasa >= secilebilirEczaneler.lalapasa.length) indexes.lalapasa = 0
    if (indexes.suloglu >= secilebilirEczaneler.suloglu.length) indexes.suloglu = 0
    if (indexes.ipsala >= secilebilirEczaneler.ipsala.length) indexes.ipsala = 0
    if (indexes.uzunkopru >= secilebilirEczaneler.uzunkopru.length) indexes.uzunkopru = 0
  }
  setEczaneler(updated_eczaneler)
}
```