# E-Öğrenme Portalı

•••

Caner Akın - 1191602618

### **Uygulamanın Ekranları**

- Giriş Ekranı

- Öğrenci Ekranı

- Yönetici Ekranı

### Uygulamanın Temel Özellikleri

#### Giriş Ekranında:

-Kişiye özel tanımlanmış kullanıcı numarası ve şifre ile giriş özelliği.

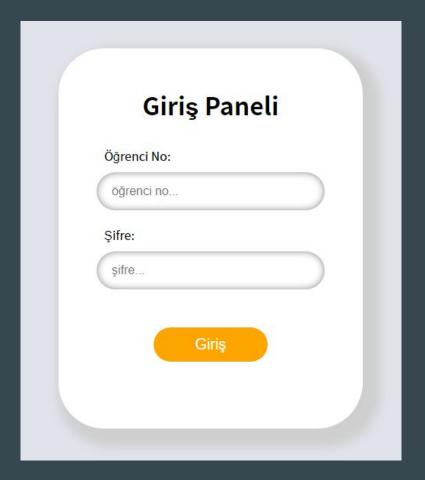
#### Öğrenci Ekranında:

- -Giriş yapan öğrenci sadece kendine önceden atanmış dersleri görebilir.
- -Yönetici tarafından yüklenmiş ders içeriğini görüntüleyebilir ve indirebilir.
- -Her derse ait yoklamaya katılabilir.

#### Yönetici Ekranında:

- -Öğrencilere ders ekleyebilir veya kaldırabilir.
- -Derslere ait yoklamayı görüntüleyebilir.
- -Ders içeriği (pdf) yükleyebilir.

## Giriş Ekranı



#### Giriş Ekranı - Kullanıcı Kontrolü (LoginCheck)

"useState" kullanarak kullanıcı numarasını ve şifresini alıyoruz.

"LoginCheck" kısmında kullanıcı numarası ve şifrenin doğruluğunu kontrol ediyoruz. Eğer "LogUser" giriş bilgilerine eşit ise TRUE döndürüyor. Giriş bilgileri uyuşmuyor ise FALSE döndürüyor, uygulama hata veriyor ve giriş yapılamıyor.

```
const [enteredUserno, setEnteredUserno] = useState('');
const [enteredPassword, setEnteredPassword] = useState('');
let LogUser= null;
const LoginCheck = (props) => {
  for (var index = 0; index < Users.length; index++) {</pre>
    if (
      props[0] === Users[index].userno &&
      props[1] === Users[index].password
      LogUser = Users[index];
      return true:
  LogUser = null;
  alert("Hatali Giris!");
  return false;
};
```

#### Giriş Ekranı - Kullanıcı Kontrolü (LoginCheck)

"LoginCheck" kısmında fonksiyon TRUE

değeri döndürdükten sonra "LogState" değerine bakılıyor. "LogState" FALSE ise uygulama Login ekranında kalıyor, TRUE ise ekrana "MainScreen" getiriliyor.

"MainScreen"e giriş yapılıp yapılmayacağı "LogState" durumuna göre karar veriliyor.

```
const [logState, setLogState] = useState(false);
const loginHandler = () => {
    setLogState(LoginCheck([enteredUserno,enteredPassword]));
};

return logState === false ? (

):(
    <mainScreen exitClick={exitHandler}></mainScreen>
);
```

#### MainScreen



#### MainScreen - Ekran Seçimi

"MainScreen" de Giriş yapılan bilgiler kontrol ediliyor. "LogUser.id=0" ise ekrana "AdminPanel" getiriliyor. 0 dan farklı ise ekrana "LessonsScreen" getiriliyor.

#### MainScreen - Öğrenci Ders Butonları

"MainScreen" de öğrencilerin bilgilerinin tutulduğu listede sahip olduğu derslere göre ekranda bu dersler buton olarak gösteriliyor.

```
const butonlar = [];
LogUser.lessons.forEach((kod) => {
 if (kod === "d1") {
   butonlar[0] = <button onClick={() => ClickHandler("d1")}>MAT101</button>;
  } else if (kod === "d2") {
   butonlar[1] = <button onClick={() => ClickHandler("d2")}>FIZ102</button>;
   else if (kod === "d3") {
   butonlar[2] = <button onClick={() => ClickHandler("d3")}>ELK321</button>;
  } else if (kod === "d4") {
   butonlar[3] = <button onClick={() => ClickHandler("d4")}>BILYD</button>;
  } else if (kod === "d5") {
   butonlar[4] = <button onClick={() => ClickHandler("d5")}>BIL3203</button>;
  } else if (kod === "d6") {
   butonlar[5] = <button onClick={() => ClickHandler("d6")}>BIL323</button>;
  } else if (kod === "d7") {
   butonlar[6] = <button onClick={() => ClickHandler("d7")}>BIL3201</button>;
   else if (kod === "d8") {
   butonlar[7] = <button onClick={() => ClickHandler("d8")}>BIL322</button>;
```

#### MainScreen - Ekran Seçimi

Tıklanılan tuşa göre "ClickHandler" ile ders bilgileri getiriliyor.

"ClickHandler"a verilen ders idsine göre "screenSelectore" ilgili "LessonsScreen" atanıyor. Ardından "screenSelector" bu veriyi ekrana gösteriyor.

```
const MainScreen = (props) => {
  const [screenSelector, setSelector] = useState();
```

### LessonsScreen - İçeriği

Ekranda "LessonScreen" gösterildiğinde,

Seçilen dersin adı,

Yoklama butonu,

Yönetici tarafından yüklenmiş ders içeriği.

#### LessonsScreen



#### LessonsScreen - Yoklamaya Katılma

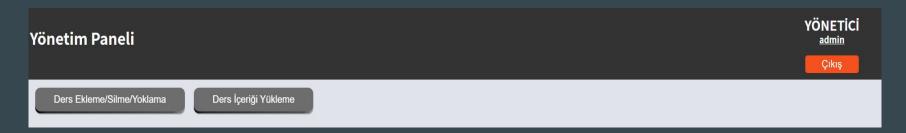
"LessonsScreen" ekranında yoklama butonuna tıklanıldığında, öğrencinin id'si yoklama tablosunda yok ise öğrenciyi kayıt eder. Eğer öğrenci var ise kayıt etmez.

# AdminPanel - İçeriği

Ekrana "AdminPanel" gösterildiğinde,

Ders Ekleme/Silme/Yoklama ekranı,

Ders içeriği yükleme ekranı.



#### AdminPanel - Ders Ekleme/Silme/Yoklama

Ders Ekleme fonksiyonu,

İlk döngüde seçilen ders, zaten öğrencinin sahip olduğu dersleri arasında ise öğrenci bu derse sahip uyarısı veriyor.

İkinci döngüde eğer ki seçilen ders öğrencide yok ise ders öğrenciye ekleniyor.

```
x = userId y= LessonsId
const Ekleme = (x, y) => {

var ax;
for (var b = 0; b < Users[x].lessons.length; b++) {

   if (Users[x].lessons[b] == "d" + (parseInt(y) + 1)) {
      alert("Öğrenci bu derse sahip");
      ax = b;
   }

if (Users[x].lessons[ax] != "d" + (parseInt(y) + 1)) {
   Users[x].lessons.push("d" + (parseInt(y) + 1));
   setMessage(<div className="texts-css">Ders başarıyla eklendi <</div>);
```

Ders İçeriği Yükleme

Yoklama Sayısı: 0 Fizik 

Öğrenciyi Derse Ekle Öğrenciyi Derse Ekle Öğrenciyi Dersten Kaldır

#### AdminPanel - Ders Ekleme/Silme/Yoklama

Ders Silme fonksiyonu,

"oldList": Dersin silinmemiş halini saklıyor.

"newList": Dersin silinmiş halini saklıyor.

"newList"e , "oldList"den silinecek ders harici bütün dersleri ekler.

"NewListi" öğrencinin yeni listesi yapar.

Bu şekilde öğrenciden istenilen ders silinmiş olur.

```
const sil = (x, y) \Rightarrow \{
 let oldList = [];
 let newList = [];
 Users.forEach((eleman) => {
   if (eleman.id == x) {
     oldList = eleman.lessons;
 for (let index = 0; index < oldList.length; index++) {</pre>
   if (oldList[index] !== "d" + (parseInt(y) + 1)) {
     newList.push(oldList[index]);
      setMessage(<div className="texts-css"> Ders basarıyla silindi X </div>);
 Users.forEach((eleman) => {
   if (eleman.id == x) {
     eleman.lessons = newList;
```

#### AdminPanel - Ders Ekleme/Silme/Yoklama

Yoklamayı görme fonksiyonu,

Yoklama alanında ki ilgili derse ait "atdlist" alanının uzunluğunu yoklamaCopy değişkenine atadıktan sonra bu değişkenin uzunluğunu "useState" ile yoklamaSayi değişkenine atayıp, bu değişkeni ekranda göstererek yoklama sayısını göstermiş oluruz.

```
let leId;
const setLesId1 = (event) => {
    leId = event.target.value;
    if(leId !== "");
    const yoklamaCopy = Yoklama[parseInt(leId)].atdList;
    console.log(yoklamaCopy.length);
    setYoklamaSayi("Yoklama Sayısı: " + yoklamaCopy.length);
```

```
export const Yoklama = [
    { id: 0, lesid: 0, atdList: [] },
    { id: 1, lesid: 1, atdList: [] },
    { id: 2, lesid: 2, atdList: [] },
    { id: 3, lesid: 3, atdList: [] },
    { id: 4, lesid: 4, atdList: [] },
    { id: 5, lesid: 5, atdList: [] },
    { id: 6, lesid: 6, atdList: [] },
    { id: 7, lesid: 7, atdList: [] },
};
```

# AdminPanel - Ders İçeriği Yükleme

