

**ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ**



BLM 4538 IOS ile Mobil Uygulama Geliştirme II

Gamze Durunesil

20290248

Proje Konusu: Movie Application

Proje Adı: MovieSea

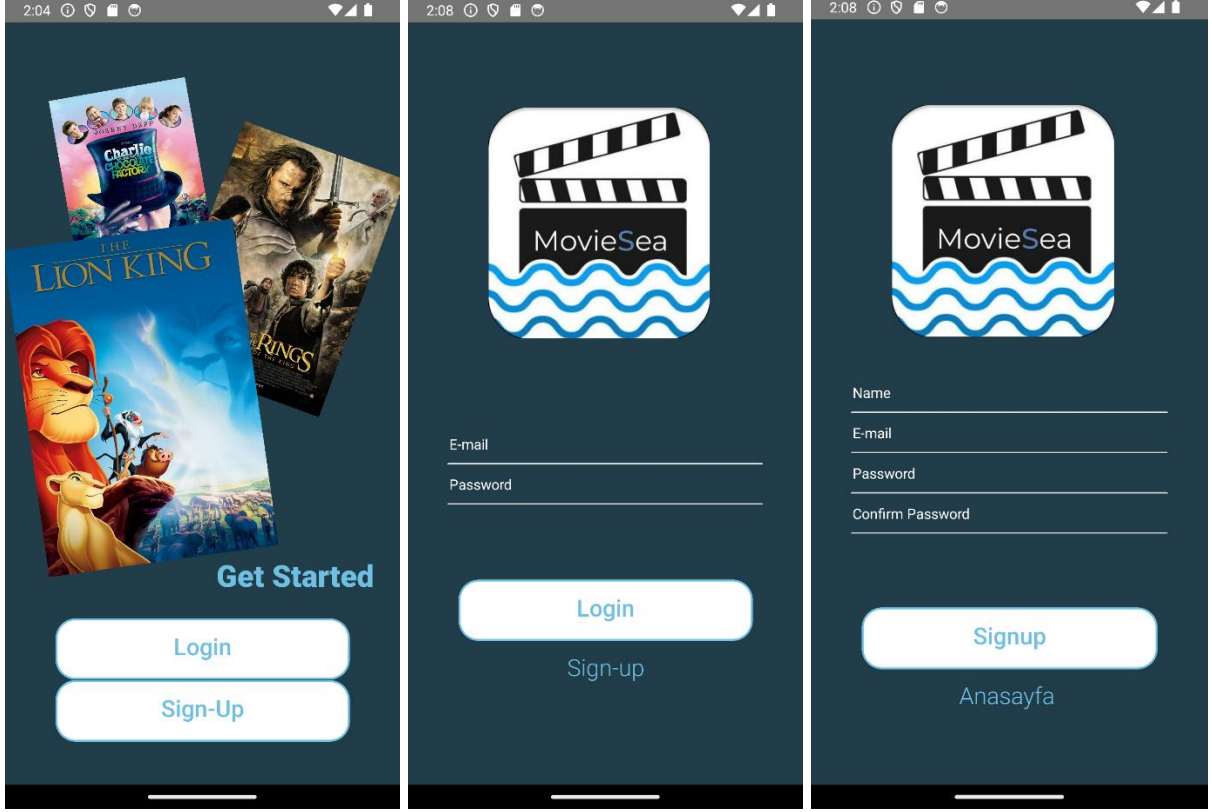
Github Linki: <https://github.com/GamzeDurunesil/BLM-4538-IOS-2>

İÇİNDEKİLER

1. KULLANICI GİRİŞİ
2. ANASAYFA
3. ARAMA SAYFASI
4. İZLEME LİSTESİ
5. VERİLERİN API İLE ÇEKİLMESİ
6. ÇIKIŞ

1. KULLANICI GİRİŞİ

Uygulama ilk açıldığında karşımızda giriş yapmak ve kayıt olmak için seçeneklerimizin bulunduğu bir ana ekranımız var.



Sign-up ve Login işlemleri için kimlik doğrulama işlemini Firebase kullanarak yaptım. Firebase'i react native dosyasına configure ettikten sonra yazdığım kodlara gerekli fonksiyonları kullanacağım eklentileri import ettim.

Kullanıcıyı kaydetmek için kullanılan kod:

```
const handleSignUp = () => {  
  createUserWithEmailAndPassword(auth, email, password)  
    .then(userCredentials => {  
      const user = userCredentials.user;  
      console.log(user.email);  
    })  
    .catch(error => alert(error.message))  
}
```

Kayıtlı kullanıcının uygulamaya girişı için kullanılan kod:

```
const handleLogin = () => {  
  
  signInWithEmailAndPassword(auth, email,password)  
  .then(userCredentials =>{  
    const user = userCredentials.user ;  
    console.log('Login with: ', user.email);  
  })  
  .catch(error => alert(error.message))  
  
}
```

Firebase'de kullanıcıların kayıt altına alınması:

Search by email address, phone number, or user UID					Add user	↺	⋮
Identifier	Providers	Created ↓	Signed In	User UID			
g@hotmail.com	✉	May 20, 2024	May 27, 2024	lzKU9Bb0onMyqVYDq0FJKnM...			
					Rows per page:	50 ▼	1 – 1 of 1 < >

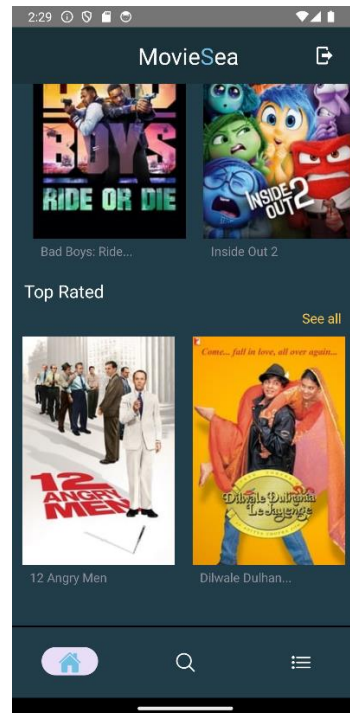
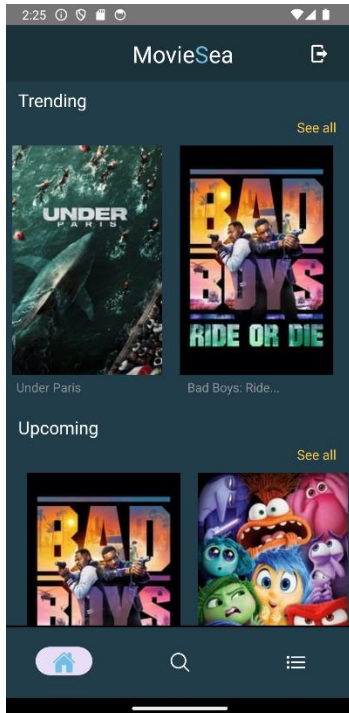
2. ANASAYFA

Uygulamaya giriş yaptıktan sonra bizi bir anasayfa karşılıyor. Bu anasayfaya ve kullanacağımız diğer sayfalara kolayca ulaşabilmek için Bottom Tab Bar kullandım.

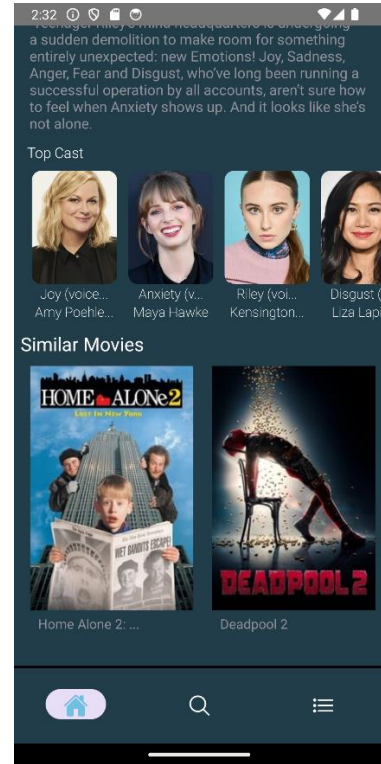
Bottom tab bar'da sayfalar arası geçişi sağlayan kod:

```
tabBarIcon: ({focused,color,size}) => {  
  let iconName;  
  let rn = route.name;  
  
  if(rn === homename){  
    iconName = focused ? 'home' : 'home-outline'  
  }else if (rn === searchname) {  
    iconName = focused ? 'search' : 'search-outline'  
  }else if (rn === watchname) {  
    iconName = focused ? 'list' : 'list-outline'  
  }  
  return <Ionicons name={iconName} size={26} color={color} />  
},  
),  
<Tab.Screen name={homename} component={ScreenStack} />  
<Tab.Screen name={searchname} component={SearchStack}/>  
<Tab.Screen name={watchname} component={WatchList}/>
```

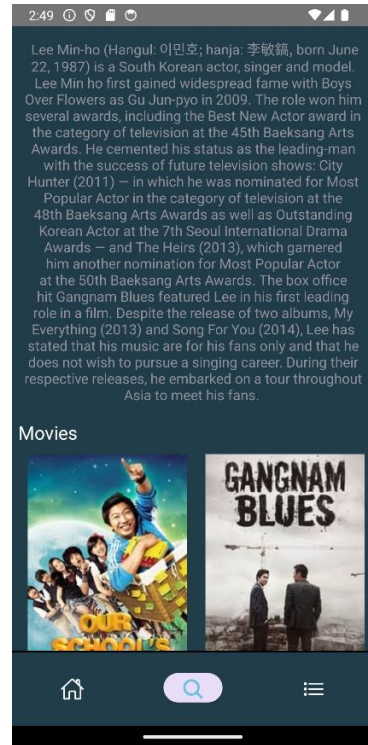
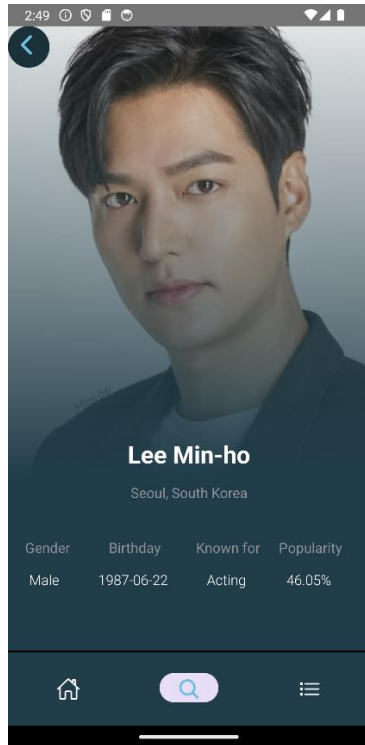
Anasayfada güncel olarak bulunan trend, gelecek ve popüler olan filmleri yana kaydırmalı bir şekilde görüyoruz.



Filmlere tıkladığımızda ise filmin detay sayfası ile karşılaşırız. Detay sayfasında ise filmin türü, yapım yılı, oyuncularını ve baktığımız filme benzeyen diğer filmleri görürüz.

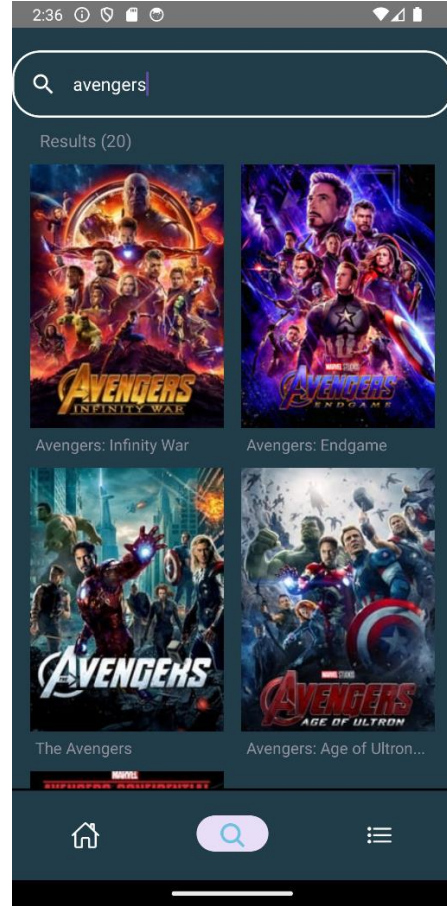


Oyuncu detaylarına bakmak istersek, oyuncuya tıkladığımızda oyuncunun doğduğu yeri, tarihi, biyografisini ve filmlerini görürüz.



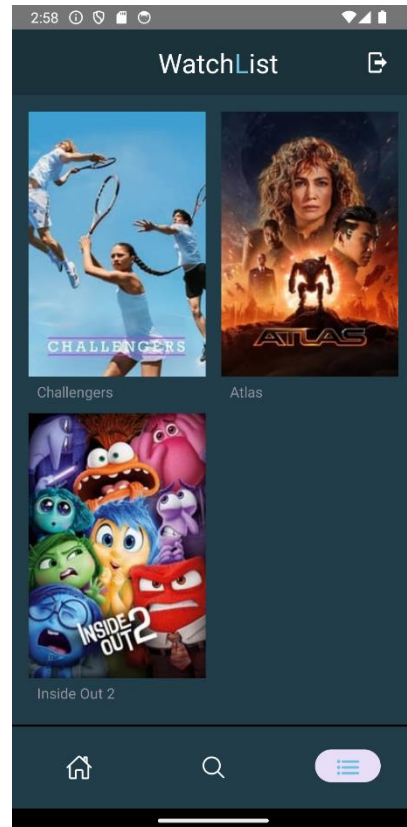
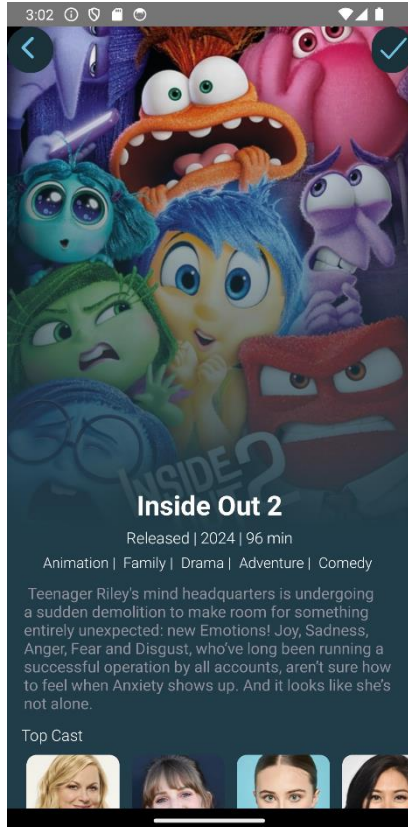
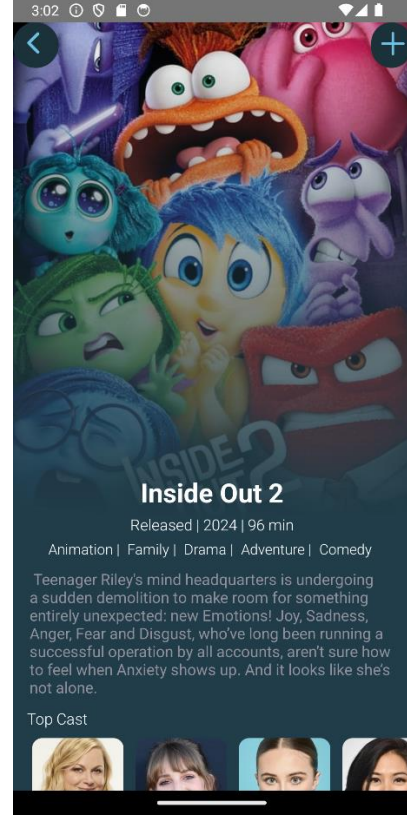
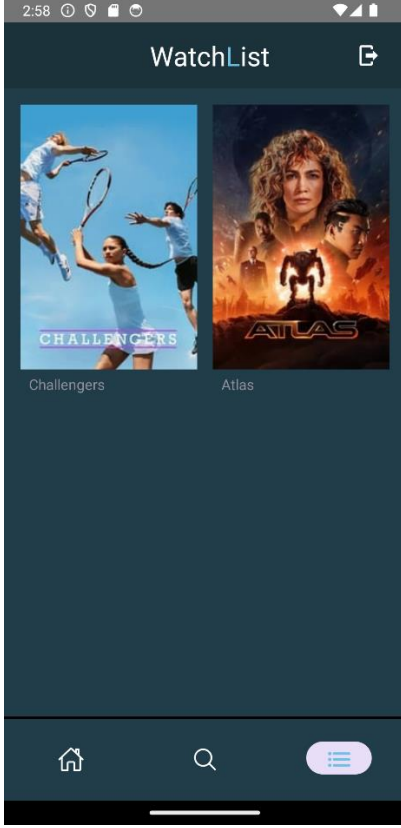
3. ARAMA SAYFASI

Arama sayfasına geldiğimizde ilk önce search bar var. Search bar aramaları harfi harfine bakarak yapıyor. Arama yapmadığımızda hiçbir seçenek gösterilmiyor. Arama yaptığımızda ise uygun filmlerin hepsini görebiliyoruz. Bu filmlere tıkladığımızda ise film detaylarına erişiyoruz. Filmin detaylarından ise oyuncu ya da benzer filmlerden istediğimize bakabiliriz. Bunların görünümü film ve oyuncu detay sayfası ile aynı.



4. İZLEME LİSTESİ

Film detay sayfasında bulunan sağ üstte yer alan + ikonuna basarak istediğimiz filmleri izleme listesine ekleyebiliriz. Eklediğimizde ikonumuz ✓ şeklinde olacaktır. Eklediğimiz filmleri izleme listesinden görebilir ve detay sayfasına gidebiliriz.



Filmleri izleme listesine eklemek için Firebase Firestore Database kullandım. Sağ üstteki ikona tıkladığımızda filmin id, isim ve poster yolu bilgileri watchlist adlı veri tabanına ekleniyor. Daha sonra veri tabanında bulunan bilgileri izleme listesi sayfasında görüntülemek için çekiyoruz.

Verilerin “watchlist” veri tabanına kaydedilmesi:

```
const addToWatchlist = async () => {  
  try {  
    await addDoc(collection(db, "watchlist"), {  
      id: item.id,  
      name: item.title,  
      poster: item.poster_path,  
    });  
  }  
};
```

Verilerin çekilmesi:

```
const getData = async () => {  
  try {  
    const querySnapshot = await getDocs(collection(db, "watchlist"));  
    const dataList = [];  
    querySnapshot.forEach((doc) => {  
      dataList.push(doc.data());  
    });  
    setData(dataList);  
  }  
};
```





Verilerin izleme listesi sayfasında gösterimi:

```
{data.map((item, index) => {
  return(
    <TouchableWithoutFeedback
      key={index}
      onPress={()=> navigation.push('MovieDetails', item)} >
      <View style={{padding:1,marginBottom:4, }}>

        <Image
          source={{uri: image185(item?.poster)}}
          //source={require("../assets/lotr.jpg")}
          style={styles.photos}/>

        <Text style={styles.nameline}> {item.name.length>14? item.name.slice(0,14)+'...': item.name} </Text>
      </View>
    </TouchableWithoutFeedback>
  )
})}
```

Watchlist adlı veri tabanı ve verilerin gösterimi:

 (default)	 watchlist	 ibgmUgpCenngLMQXzNcF
+ Start collection	+ Add document	+ Start collection
watchlist >	<div>RYnG7dF0H0Y2AReUd7M</div> <div>h6hMA0qkYQXmFkAFk4pq</div> <div> ibgmUgpCenngLMQXzNcF ></div>	<div>+ Add field</div> <div>id: 1022789</div> <div>name: "Inside Out 2"</div> <div>poster: "/vpnVM9B6NMmQpWeZvzLvDESb2QY.jpg"</div>

5. VERİLERİN API İLE ÇEKİLMESİ

Film verilerini The Movie Database (TMDb) hazır veri tabanından API kullanarak çektim. İlk önce çekeceğimiz veriler için sayfa linki oluşturuyoruz.

Fimlerin gösterimi için link:

```
const apiBaseUrl = 'https://api.themoviedb.org/3'

const trendingMoviesEndpoint = `${apiBaseUrl}/trending/movie/day?api_key=${apiKey}`
const upcomingMoviesEndpoint = `${apiBaseUrl}/movie/upcoming?api_key=${apiKey}`
const topRatedMoviesEndpoint = `${apiBaseUrl}/movie/top_rated?api_key=${apiKey}`
const searchMoviesEndpoint = `${apiBaseUrl}/search/movie?api_key=${apiKey}`
```

Benzer filmlerin, filmlerin ve oyuncuların detayları için link:

```
const movieDetailsEndpoint = id => `${apiBaseUrl}/movie/${id}?api_key=${apiKey}`
const movieCreditsEndpoint = id => `${apiBaseUrl}/movie/${id}/credits?api_key=${apiKey}`
const similarMoviesEndpoint = id => `${apiBaseUrl}/movie/${id}/similar?api_key=${apiKey}`
const personDetailsEndpoints = id => `${apiBaseUrl}/person/${id}?api_key=${apiKey}`
const personMoviesEndpoints = id => `${apiBaseUrl}/person/${id}/movie_credits?api_key=${apiKey}`
```

Film posterleri ve oyuncuların fotoğrafları için link:

```
export const image500 = path=> path? `https://image.tmdb.org/t/p/w500${path}` : null;
export const image342 = path=> path? `https://image.tmdb.org/t/p/w342${path}` : null;
export const image185 = path=> path? `https://image.tmdb.org/t/p/w185${path}` : null;
```

Daha sonra verileri çekmek için fonksiyonumuzu yazıyoruz.

```
const apiCall = async (endpoint, params) => {
  const options = {
    method: 'GET',
    url: endpoint,
    params: params? params: {}
  }
  try {
    const response = await axios.request(options);
    return response.data;
  } catch (error) {
    console.log('error:', error);
    return {}
  }
}
```

Bu fonksiyonumuza yukarıdaki linkleri parametre olarak atayarak filmlerin verisini çekiyoruz.

```
export const fetchTrendingMovies = ()=>{  
  return apiCall(trendingMoviesEndpoint);  
}
```

Filmlerin id'lerini kullanarak bu id üzerinden filmlerin, oyuncuların ve benzer filmlerin detay verilerini çekiyoruz.

```
export const fetchMovieDetails = id => {  
  return apiCall(movieDetailsEndpoint(id));  
}  
export const fetchSimilarMovies = id => {  
  return apiCall(similarMoviesEndpoint(id));  
}  
export const fetchPersonDetails = id => {  
  return apiCall(personDetailsEndpoints(id));  
}
```

Daha sonra çektiğimiz verileri göstermek istediğimiz sayfanın kodunda bu verilere ulaşım sağlıyoruz.

```
const getMovieDetails = async id =>{  
  const data = await fetchMovieDetails(id);  
  //console.log('got movie details: ',data);  
  if(data) setMovie(data);  
}
```

Ulaştığımız film posteri, adı, film detayı, oyuncularını aşağıdaki koddaki yöntemle aynı şekilde ekranda gösteriyoruz.

```
data.map((item, index)=>{  
  return(  
    <TouchableWithoutFeedback  
      key={index}  
      onPress={()=> navigation.push('MovieDetails', item)} >  
      <View style={{padding:10,}}>  
  
        <Image  
          source={{uri: image185(item.poster_path)}}  
          style={styles.photos}/>  
        <Text style={styles.nameline}> {item.title.length>14? item.title.slice(0,14)+'...': item.title} </Text>  
  
      </View>  
    </TouchableWithoutFeedback>  
  )  
})
```

6. ÇIKIŞ

Anasayfa ve izleme listesi sayfalarında sağ üstte bulunan logout butonu kullanarak uygulamamızdan çıkış yapabiliriz. Çıkış yaptığımızda bizi giriş sayfasına yönlendirecektir.

Çıkış işlemi için kullanılan kod:

```
const handleLogout = () =>{  
  
  signOut(auth)  
  .then(()=>{  
    | navigation.replace("Welcome")  
  })  
  .catch(error => alert(error.message))  
}
```