**NETFLİX PLATFORMU PROJESİ**

Derya ÖZGÜ, Mehmet KIYAK

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[ozguderya@gmail](mailto:ozguderya@gmail).com, [ben@](mailto:ben@)mehmetkiyak.com.tr

Özet

*Projede mysql veri tabanı kullanılarak netflix uygulaması yapılmıştır. Proje kullanıcı kaydı, kullanıcı girişi, anasayfa, program sayfası ve program izleme sayfalarından oluşmaktadır.*

*Netflix, uzu metrajlı filmler, belgeseller, diziler, programlar ve animeler izleme platformudur.*

# Giriş

Program ilk çalışma esnasında giriş yap ekranı açılarak başlatılmaktadır. Kullanıcı bilgileriniz yok ise kayıt ol butonuna tıklayarak kullanıcı kaydı oluşturulabilmektedir. Kullanıcı kayıt sayfasından giriş sayfasına geri dönüş yapılarak kullanıcı bilgileriniz ile programa giriş yapılıyor. Ana sayfa da kullanıcının hızlı arama yapabilmesi için türlere göre ikonlar ve butonlar oluşturulmuştur. Ayrıca farklı isimlerde arama yapılabilmesi içinde üst alanda arama bölümü de mevcuttur.

# Temel Bilgiler

Program JAVA programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak “Netbeans IDE” kullanılmıştır. Veri tabanı olarak MYSQL kullanılmıştır. Akış diyagramı için [draw.io](http://draw.io) online diyagram yapma programı kullanılmıştır

# Tasarım

Netflix Platformu programının programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

# Algoritma

Program öncelikle giriş sayfasında kullanıcıdan alınan eposta/kullanıcı adı ve şifre bilgilerinin alınması ile başlamaktadır. Alınan bilgiler veri tabanında sorgulatılıyor eğer varlığı belirtilirse ana sayfa ara yüzüne geçiş yapılıyor. Ana sayfa da belirtilen türlerden birinin seçilmesi durumunda arama sayfasına yönlendirilerek seçilen türe ait veri tabanında arama işlemi yapılıyor bulunan veriler alt tarafta gösterilen tablo içerisine yazılıyor.

Ayrıca üs alan da bulunan arama bölümden de arama yapıldığından yine bu arama sayfası üzerinde arama gerçekleştiriliyor. Tabloda listelenen programlardan izlemek istediğiniz programı seçerek program izlemek için oluşturmuş olduğumuz ara yüz karşınıza gelecektir. Eğer daha önceden bu programı izlemiş iseniz program izleme sayfasında karşınıza bir uyarı mesajı gelecektir ve size kaldığınız yerden başlayıp başlamamanız sorusunu soracaktır. Çıkış için ise sağ üstte bulunan X butonuna tıklamanız yeterlidir.

Platformun açıklamaları şu şekilde ifade edilmektedir;

1. **Kullanıcı Giriş Ara Yüzü:** Tasarlamış olduğumuz giriş ara yüzünden sisteme giriş yapabilmesi için kullanıcıdan eposta/kullanıcı adı istenilmektedir. Eğer kullanıcı bilgisi yok ise kayıt ol butonuna tıklayarak kayıt ol için tasarladığımız ara yüze yönlendirilecektir.
2. **Kullanıcı Kayıt Ol Ara Yüzü:** Kayıt ol ara yüzünden kullanıcıdan ad soyad, eposta, doğum tarihi ve şifre istenilmektedir. Kullanıcı kaydı tamamlandıktan sonra giriş yap butonuna tıklayarak giriş ara yüzüne geri dönüş yapılabiliyor. Ayrıca kayıt ol ara yüzünden bulunan türlerden 3 tanesinin seçilmesi durumunda seçilen türlere ait en çok puan alan 2 program altta bulunan tabloda listlenmektedir.
3. **Ana Sayfa Ara Yüzü :** Bu ara yüzde kullanıcını türlere göre daha hızlı arama yapabilmesi amaçlanmıştır. Seçilen türe ait arama ara yüzüne geçilerek programlar listelenmektedir.
4. **Program Arama Ara Yüzü:** Ana sayfa gelen türlere göre veya üst alanda arama bölümünden arama yapıldığında bu ara yüz açılacak ve alt tarafta bulunan tabloda listelenecektir. Ayrıca listeden tıklanan program içinde program izleme ara yüzü açılacaktır.
5. **Program İzleme Ara Yüzü :** Yapılan arama sonucunda açılması istenilen program bu ara yüzde açılacaktır. Bu ara yüzde seçilen programın genel bilgileri ve video alanı mevcuttur. Videoya tıkladığında videonun başlatılıyor yazısı çıkacaktır. Eğer kullanıcı programı kapatıp tekrar bu programı izlemek için bu ara yüze geldiğinde kaldığı yerden başlamak istiyor musunuz? sorusuyla karşılaşacaktır. Ayrıca sol altta bulunan puanlama alanından bu programı puanlaması istenmektedir.

# Kullanılan Sınıflar ve Fonksiyonlar

* + 1. **Database.java :** Database sınıfında veri tabanı bağlantısı için gerekli bilgiler verildi. Veri tabanı adı, kullanıcı adı, sunucu bilgisi ve port bilgilerini barındırıyor.
    2. **VeritabaniIslemleri.java :** Bu sınıfta program için veri tabanına bağlı bütün fonksiyonel işlemler yapıldı. Bu fonksiyonları açıklamak gerekirse;
* **VeriTabaniIslemleri():** Sınıfa ait constructor fonksiyonu olup veri tabanına bağlanmak için yazmış olduğumuz fonksiyondur.
* **GirisYap(String KullaniciAdi, String Parola):** GirisYap.java sınıfından gelen kullanıcı bilgilerinin veri tabanı üzerinde varlığının doğruluğu tespit edilerek true veya false değeri döndürmektedir.
* **KullaniciEkle(String AdSoyad,String Eposta,String DogumTarihi,String password) :** Kayit\_Ol.java sınıfından kullanıcı bilgilerinin fonksiyona gönderilmesi ile önce kullanıcının veri tabanında kaydının var olup olmadığını kontrol ettikten sonra kayıt işlemi yapılıyor.
* **KullaniciKontrol(String KullaniciAdi):** KullaniciEkle fonksiyonunda dediğimiz gibi kullanıcının veri tabanında daha önce kaydının varlığını kontrol etmek için bu fonksiyon ile kontrolünü yapıyoruz.
* **KullaniciBilgisi(String kullanici\_adi):** Kullanıcının ad soyad bilgisini öğrenmek için yazmış olduğumuz fonksiyondur.
* **KayitProgramListesi(int tur):** Kayıt esnasında seçilen türlere göre programları listelemek için yazılmış olduğumuz fonksiyondur.
* **ArrayList<Programlar> ProgramlariGetir():** Bufonksiyonprogramları listelemek için Programlar.java sınıfı oluşturduk bu sınıf içerisine kaydını yazdığımız programları listelememizi sağlamaktadır.
* **Turler(int id):** Programları listelerken bir programın birden fazla türe ait olmasından dolayı bu türleri listelemek için yazmış olduğumuz bu fonksiyonda arada virgülleri düzenli yazmak için TurAdeti fonksiyonundan gelen adet bilgisine göre yazılmaktadır.
* **Tur\_Adeti(int id):** Turler fonksiyonunda belirtiğimiz üzere bir programa ait kaç adet tür olduğunu bulmak için kullanıyoruz.
* **TurBul(int tur):** Tür adını bulmak için yazdığımız fonksiyondur.
* **ProgramIzlemeKontrol(String program\_id,String KullaniciAdi):** Kullanıcının programı daha önce izleyip izlemediğini kontrol etmemize yardımcı oluyor.
* **KullaniciIzlemeOlustur(String program\_id, int user\_id):** Kullanıcı programı ilk defa izliyor ise izlenme kaydını oluşturuyoruz.
* **KullaniciIzlemeGuncelle(String program\_id,int user\_id, int bolum, String sure):** Kullanıcı programı yarım bırakıp programı kapatması sonucu tekrar programa izlemeye geldiğinde nerede kaldığını ve daha sonra izlediği yere kadar olan süreyi ve bölümü güncelliyoruz.
* **UserIdBul(String Kullanici):** Kullanıcı id ‘sini öğrenmek için yazmış olduğumuz fonksiyondur.
* **ProgramBilgileri(String Program):** Kullanıcı program izleme ara yüzüne geldiğinde programa ait genel bilgileri görüntülemek için veri tabanından gelen bilgileri alıyoruz.
* **ProgramPuanla(String program\_id,String KullaniciAdi,String puan):** Kullanıcı izlediği programı puanlamak istediğinde gönderilen puan değerini veri tabanına yazan fonksiyondur.
* **ProgramPuanBul(String program\_id,String KullaniciAdi):** Kullanıcının programa ait daha önce vermiş olduğunu puanı öğreniyoruz.
  + 1. **GirisYap.java :** Giriş ara yüzünün tasarımı ve kullanıcı girişinin var olduğunu sınıftır.
    2. **Kayit\_Ol.java :** Bu sınıfta öncelikle kullanıcı tasarımı mevcuttur. Kullanıcı kaydı yapmamızı ve seçilen türe ait programları listelediğimiz sınıftır.
* **ProgramlariGoruntule():** Kayıt ol sayfasında programları listelemek için kullandığımız fonksiyondur.
  + 1. **Anasayfa.java :** Programın ana yüzünü temsil eden anasayfa sınıfında kullanıcının türe göre hızlı arama yapabilmesi için gerekli türler belirtilmiş olup seçilen türe tıklanmasıyla o türe ait programlar listelenmektedir.
    2. **Programlar.java :** Bu sınıfta veri tabanında var olan programları okuduktan sonra bu sınıf içerisine kaydını yapıyoruz ardından burada ki verileri okuyup tabloya yazıyoruz.
    3. **ProgramListesi.java :** Programlar sınıfına kaydettiğimiz verileri tabloya yazma işlemi yapıyoruz. Bu sayede kullanıcıya var olan programları listeliyoruz.
* **ProgramlariGoruntule():** Programlar sınıfından gelen verileri tabloya yazmamızı sağlayan fonksiyondur.
* **dinamikAra():** Tabloda listelenen programları gelen arama değerine göre arayıp listeleyen fonksiyondur.
  + 1. **ProgramIzle.java:** Program listesinden seçilen programın izlenmesi için tasarladığımız sınıftır. Bu sınıfta program bilgileri, program içeriği ve programı puanlama alanı mevcuttur.
* **ProgramKontrol(int islem):** Kullanıcının daha önce programı izleyip izlemediğini kontrol ettiğimiz fonksiyon olup. Programın ilk izlenme kaydını ve daha sonra izlemesini güncellediğimiz ayrıca kullanıcının kaldığı yerden devam edip etmemesin sorduğumuz fonksiyondur.
* **PuanGoster():** Kullanıcının daha önceden bu programı izleyip puan verip vermediğini kontrol edip ekrana yazan fonksiyondur.

# Sonuçlar

Programımız proje kapsamında istenilen tüm isterleri yerine getirmektedir. Gerekli bütün testler yapılmış ve doğru sonuçlar elde edilmiştir. Her hangi bir hata veya eksik mevcut değildir.

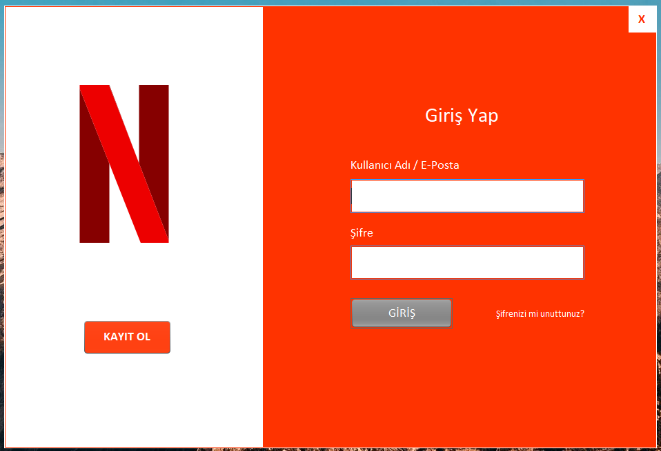
# Kaynakça

1. <https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-java-gelistirici-kursu/> kaynağından java diline ait gerekli terimler ve veri tabanı ilişkilendirilmeleri hakkında bilgi edinildi.
2. <https://www.youtube.com/watch?v=XfqV5AbCqLU&t=406s> kaynağından form tasarımı hakkında bilgi edildi.

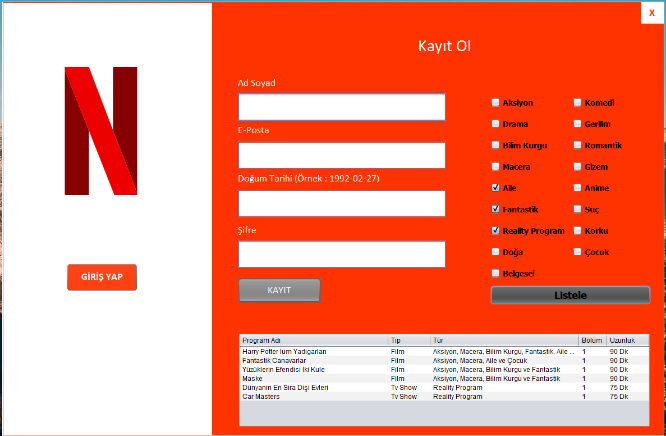
# 6. Denklemler

Program içerisinde matematiksel olarak bir hesaplama bulunmadığı üzere hiçbir denklem kullanılmamıştır.

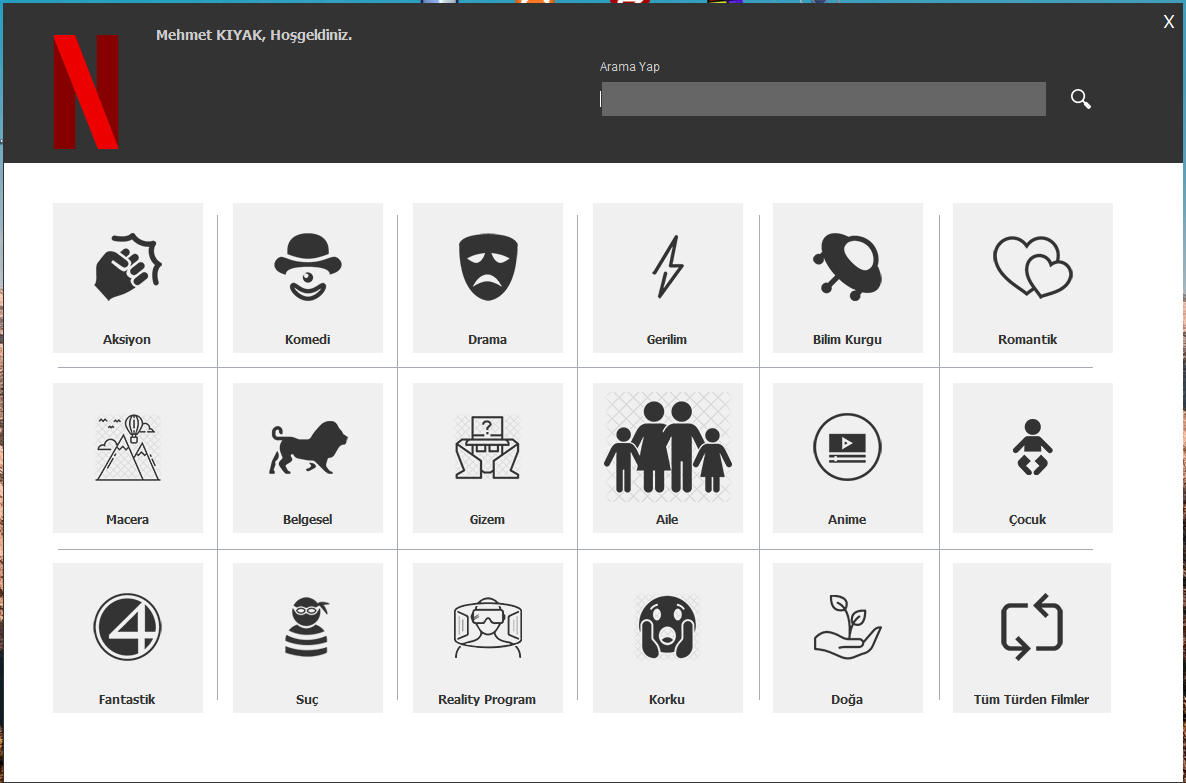
# 7. Ekran Görüntüleri



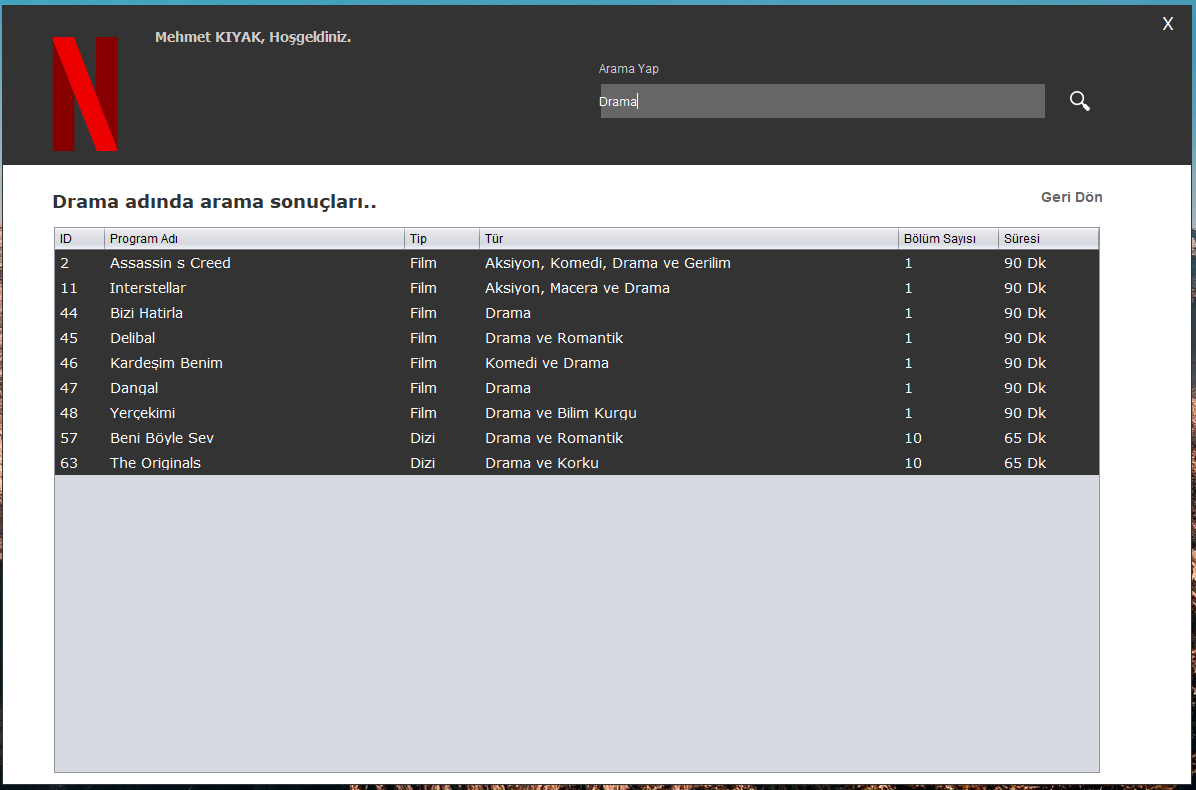
**Resim 1. Kullanıcı Girişi Ekranı**



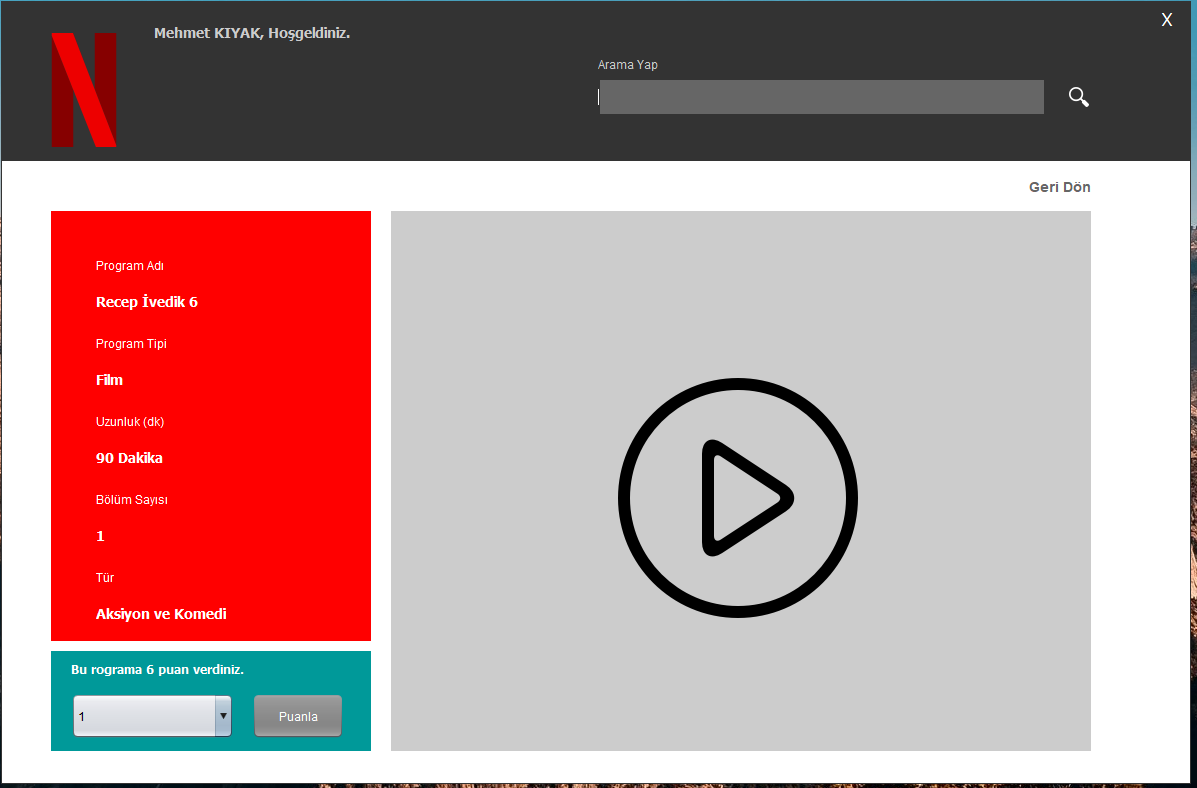
**Resim 2. Kullanıcı Kayıt Ol Ekranı**



**Resim 3. Ana Sayfa Ekranı**



**Resim 4. Program Liste Ekranı**



**Resim 5. Program İzleme Ekranı**

