****

**VERİ TABANI YÖNETİM SİSTEMLERİ DERSİ SİNEMA OTOMASYONU PROJESİ RAPORU**

**Öğretim Görevlisi**

Gülşah Tümüklü Özyer

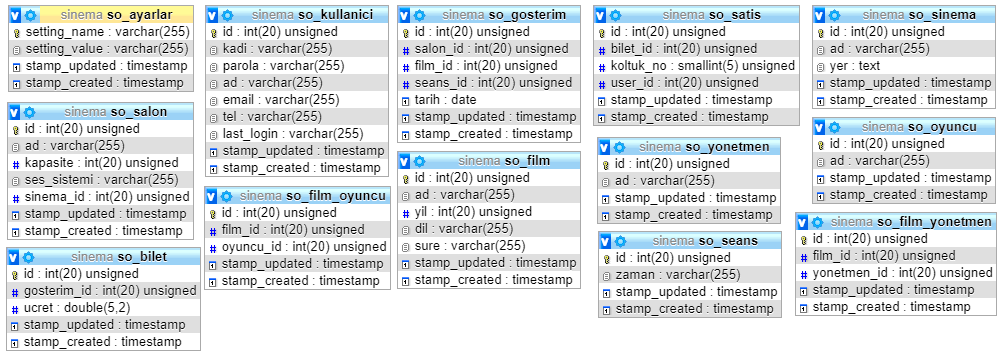
**Grup Üyeleri**

SEHER GÜREL – 150707044

KÜBRA SAĞLAM – 150707051

PHP programlama dili ve mySql sorgu dili olarak seçtiğimiz Sinema Otamasyonu projemizde Kullanıcı Girişi ve Yönetici Girişi olarak iki adet giriş ekranımız bulunmaktadır. Çalışmamızda Bootstrap da kullanılmıştır.

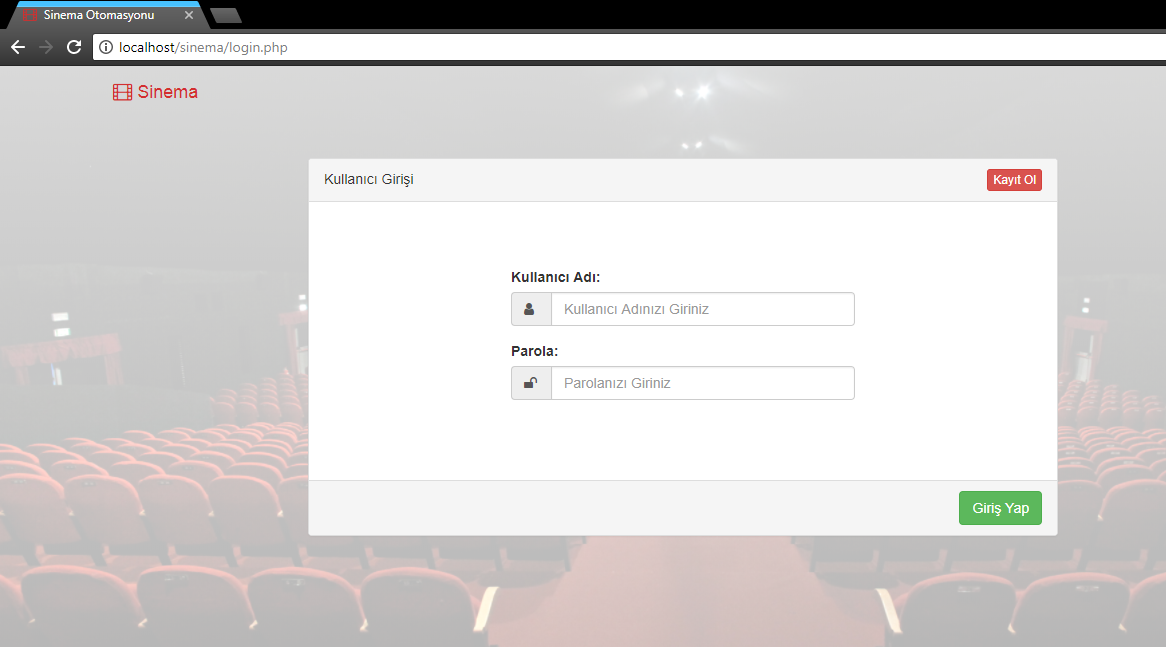
Veritabanı olarak mySql kullandığımız projemizde 13 adet tablomuz bulunmaktadır. Aşağıda tablolarımızı görüyorsunuz :



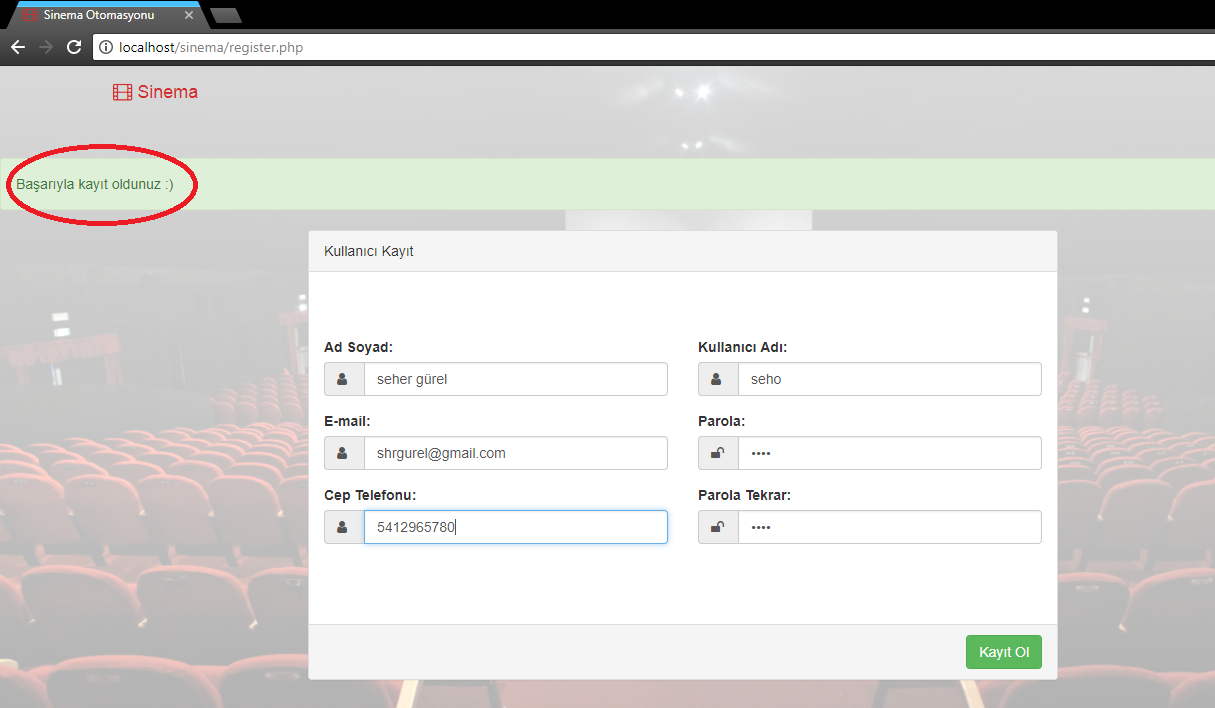
Projemizin ilk sayfasında giriş ekranı bulunmaktadır. Kullanıcı Girişi veya Yönetici Girişi olarak giriş yapabiliriz. Aşağıda Kullanıcı ve Yönetici Girişi olarak uygulayabileceğimiz adımları aşağıda sıralanmıştır.

***KULLANICI İŞLEMLERİ***

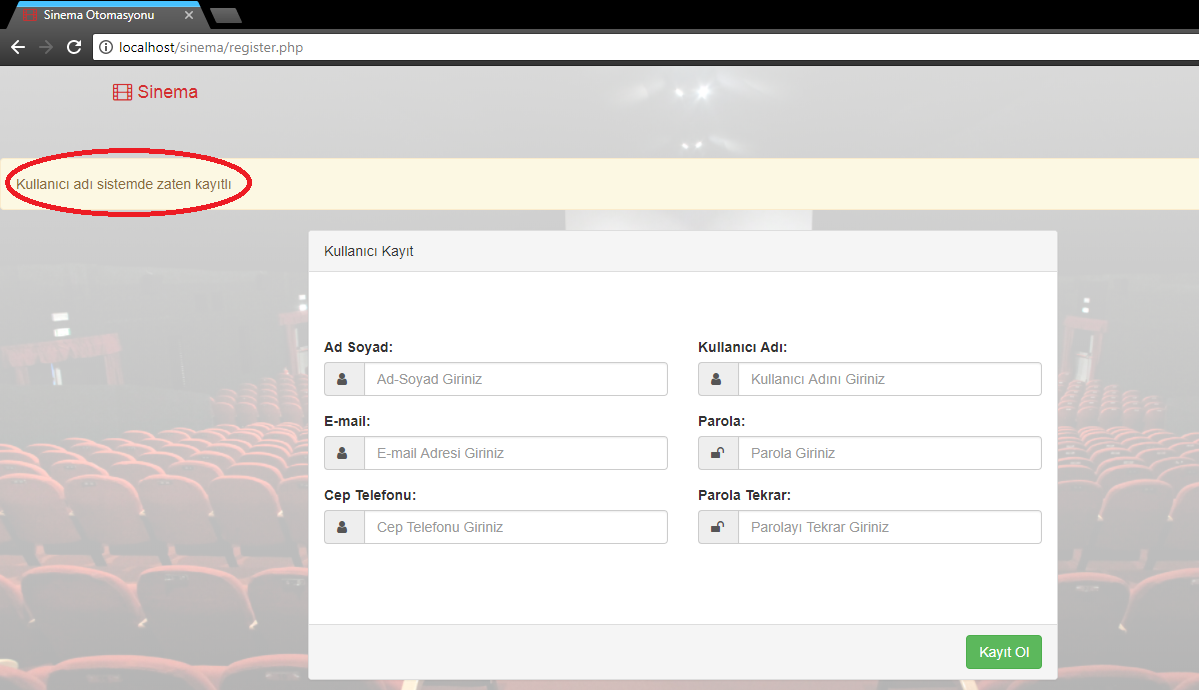
* Burada kullanıcı kayıtlı ise bilgilerini girerek sisteme giriş yapabilmektedir. Eğer kayıtlı değilse kayıt ol butonu ile kayıt yapabilmektedir.



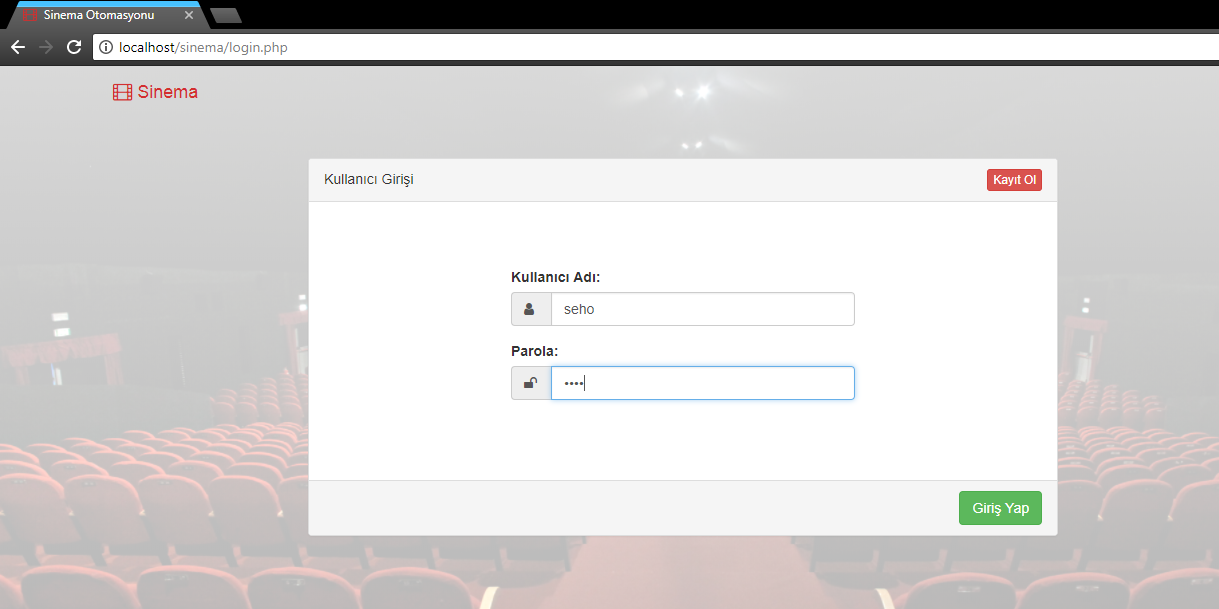
* Kullanıcının yeni kayıt yapması için kullanılan sayfamız bu şekilde görünmektedir. Verilen bilgileri doğru bir şekilde tamamlayıp kayıt ol butonuna basarak kayıt işlemini tamamlar. Daha sonra yukarıda “Kayıt işlemi başarılı” ya da “Kayıt işlemi başarısız” tarzında uyarılar vermektedir. Başarısız kayıt nedenleri eksik bilgiler ya da halıhazırda mevcut bir hesap ile kayıt yapılmak istenmesi olabilir :



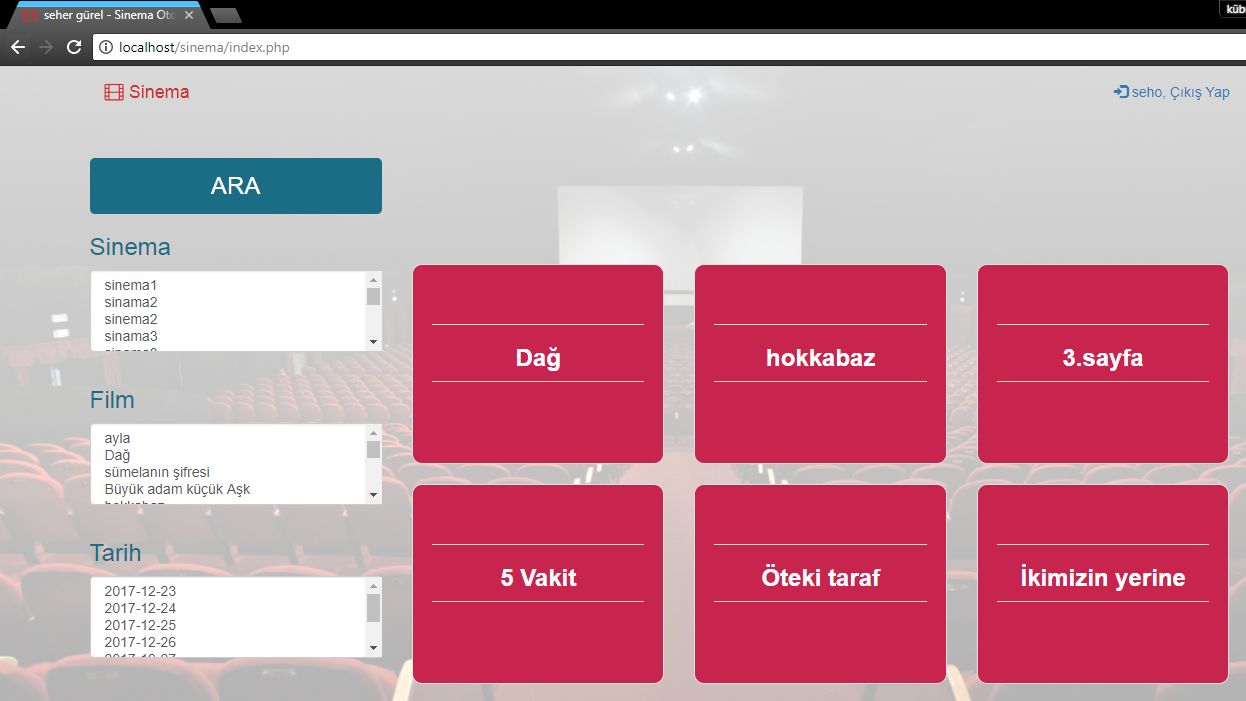
* Kayıtlı kullanıcının tekrar kayıt isteği üzerine verilen uyarı örnekteki gibidir :



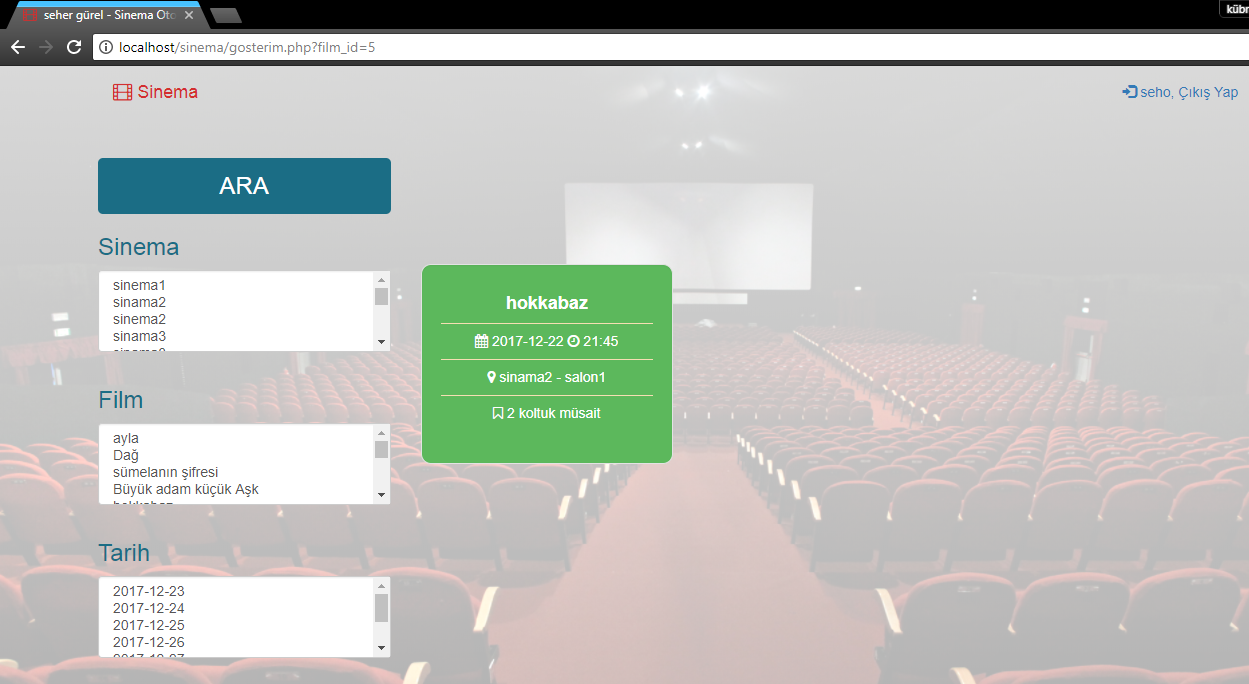
* Kayıt yaptıktan sonra artık sisteme giriş yapabiliriz. İstenilen bilgileri girdikten sonra giriş yap diyerek sistemimize giriş yapıyoruz :



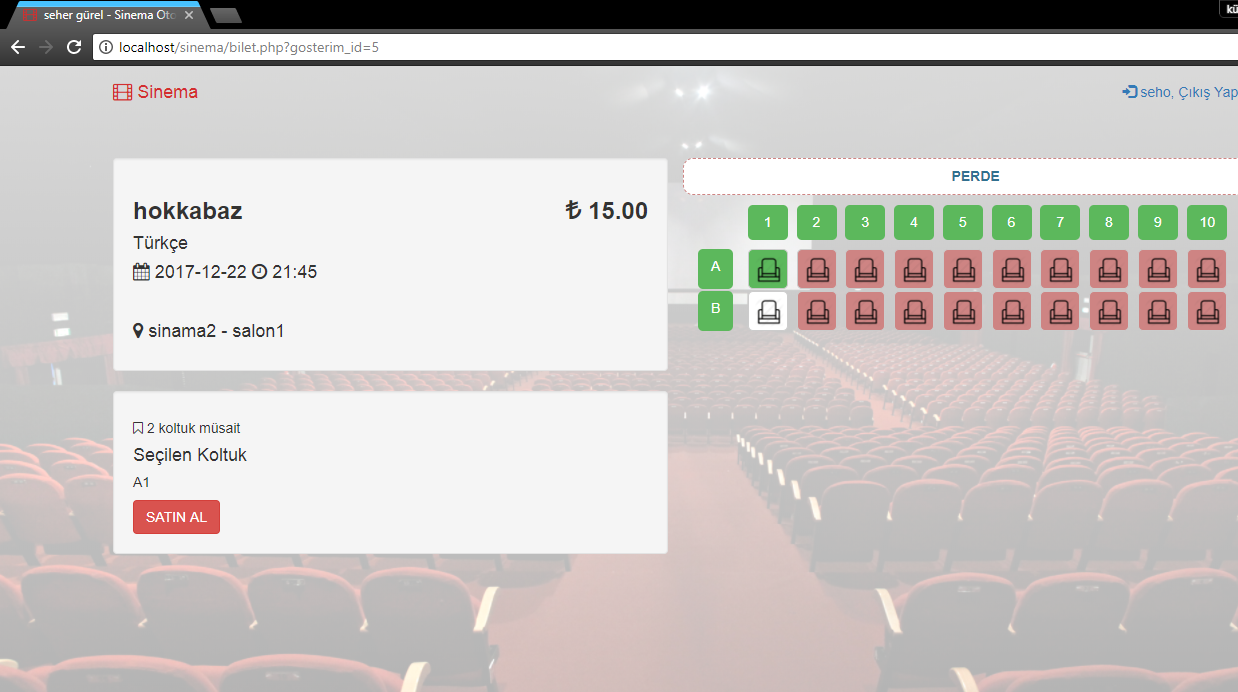
* Kullanıcı girişi ardından karşılaşacağımız sayfamız bu şekilde görünmektedir. Kullanıcı “Sinema, Film ve Tarih” seçeneklerinden dilediğini seçerek arama yapabilir veya sunulmuş mevcut arayüz ile seçimlerini gerçekleştirebilir.



* Kullanıcının “Hokkabaz” filmini seçtiğini varsayalım. Resimde de görüldüğü gibi filmin müsait koltuk sayılarını göstererek bir yönlendirme yapıyor.

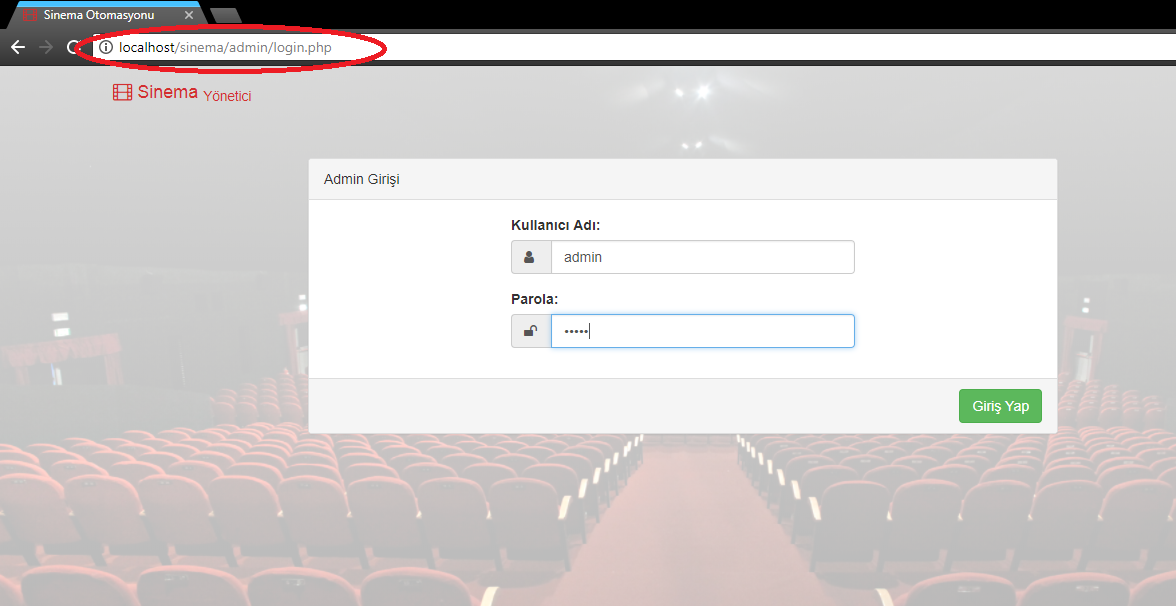


* Müsait koltuklar var ise tekrar tıklıyoruz ve istediğimiz koltuk numaralarını seçerek satın alma işlemi gerçekleştiriyoruz. Kırmızı renkli koltuklar dolu, beyaz renkli koltuklar boş ve yeşil renkte görünen de bizim seçim yaptığımız koltuk. Satın alma işleminin ardından yeşil renk kırmızı olacak.

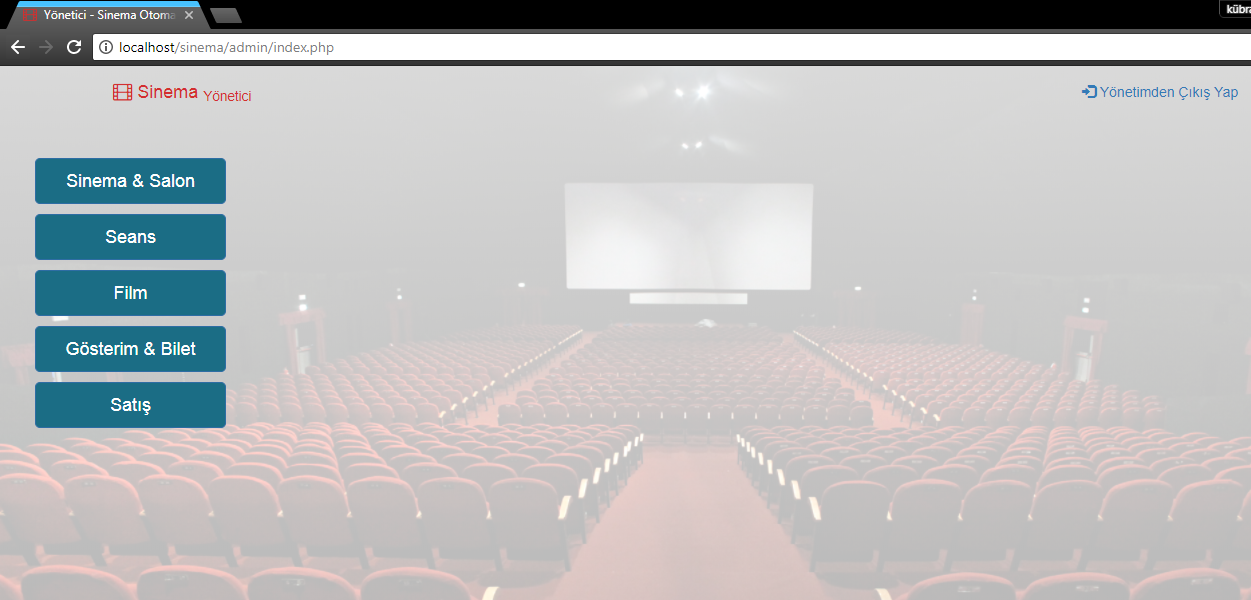


***YÖNETİCİ İŞLEMLERİ***

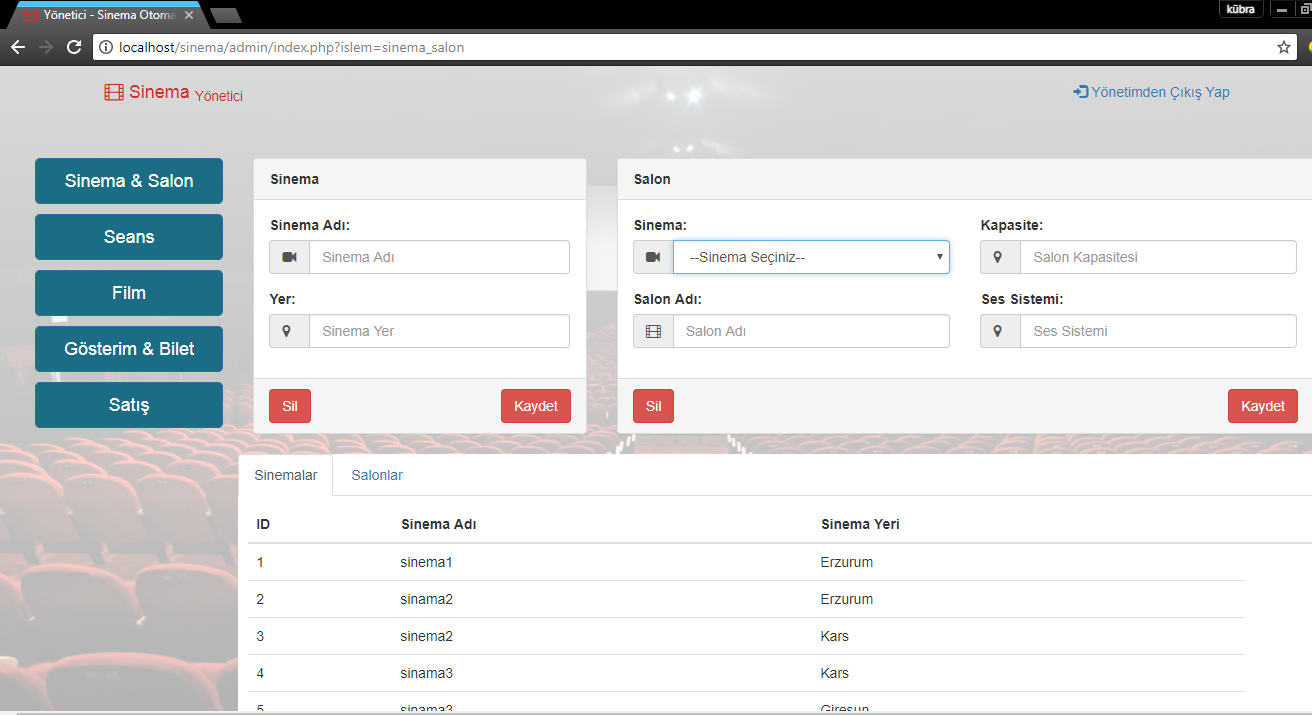
* Yönetici giriş ekranımız da müşteri giriş ekranı ile aynı özelliklere sahiptir. İstenen bilgileri girerek yönetici sayfamıza giriş yapabiliriz.



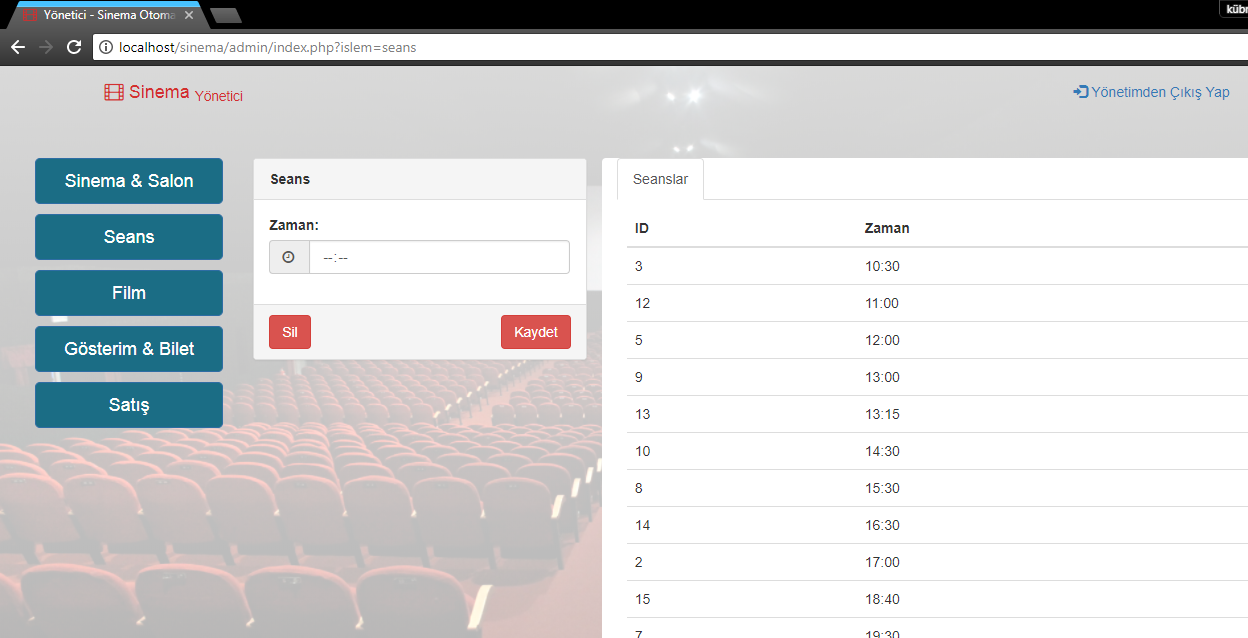
* Yönetici sayfası bize filmlerin seansları, gösterimleri, bilet işlemleri vs. hakkında bilgilerini girmemize olanak sağlıyor.



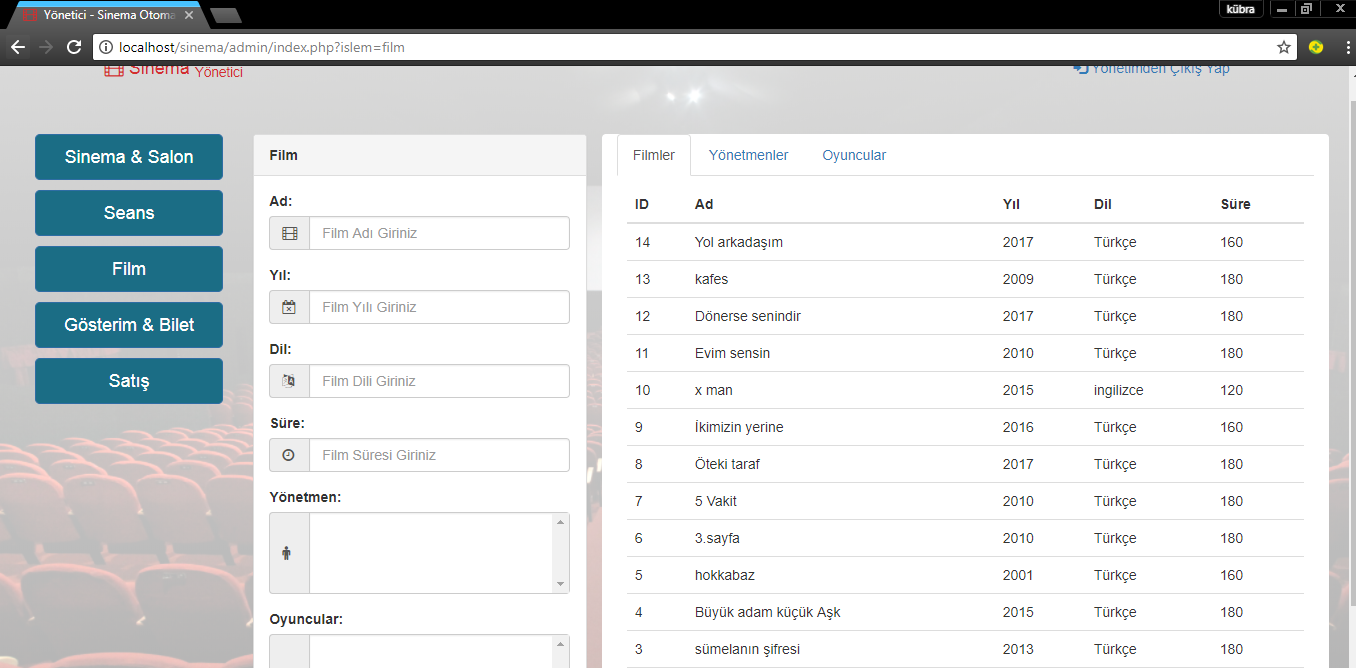
* “Sinema&Salon” sekmesine tıkladığımızda girebileceğimiz ve görebileceğimiz bilgiler :



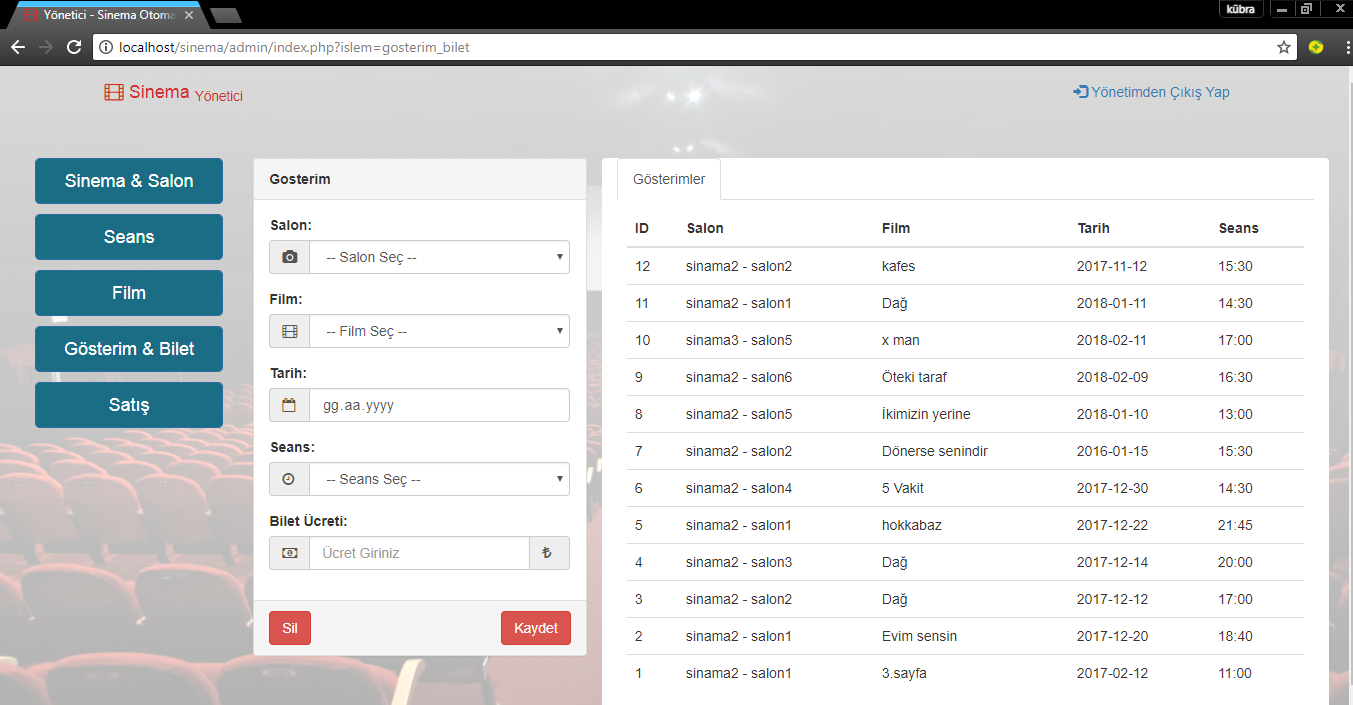
* “Seans” sekmesine tıkladığımızda girebileceğimiz ve görebileceğimiz bilgiler :



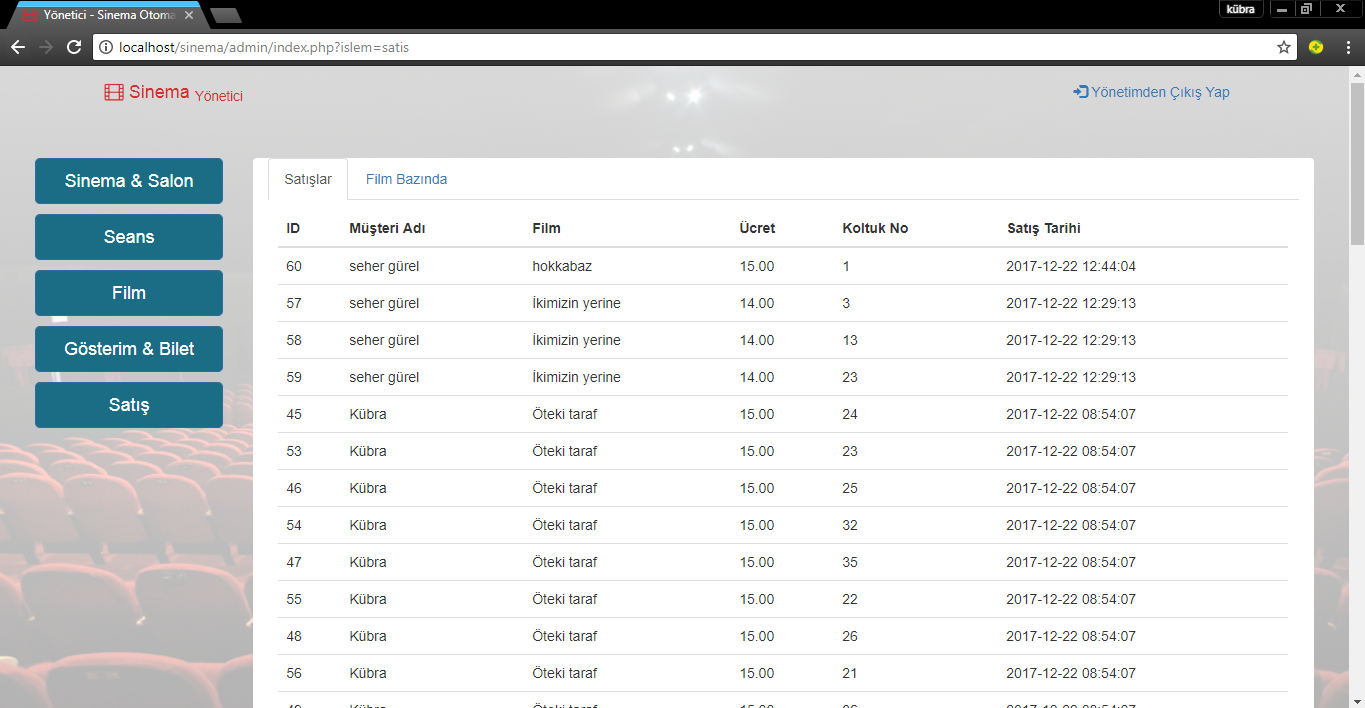
* “Film” sekmesine tıkladığımızda girebileceğimiz ve görebileceğimiz bilgiler :

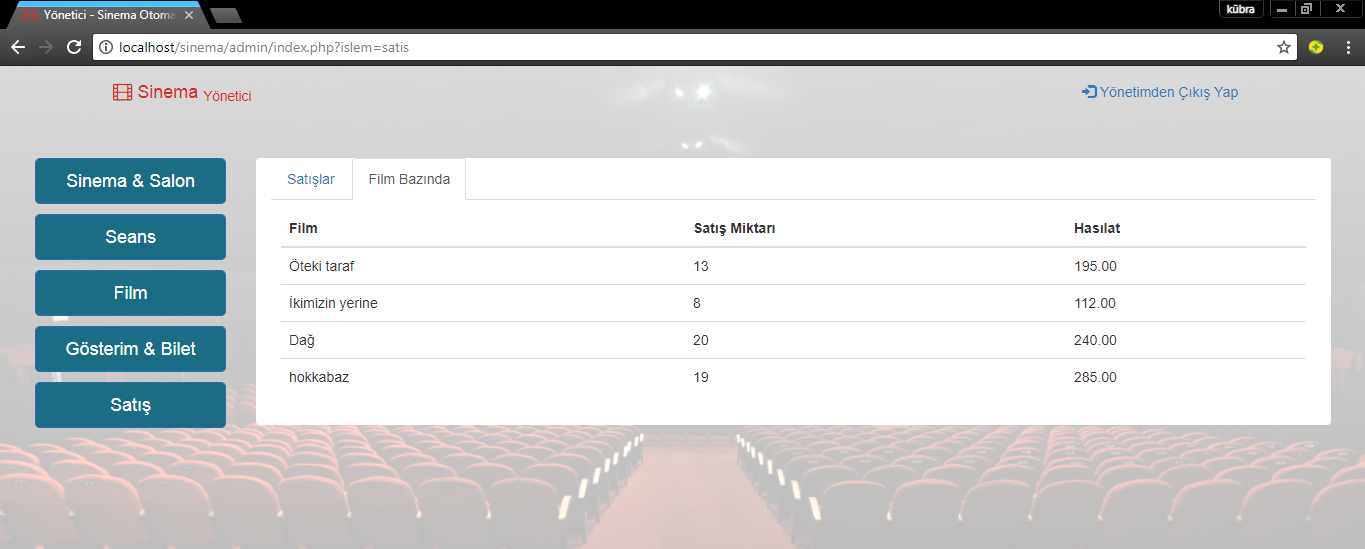


* “Gösterim&Bilet” sekmesine tıkladığımızda girebileceğimiz ve görebileceğimiz bilgiler:



* “Satış” sekmesine tıkladığımızda görebileceğimiz bilgiler :





**ÖRNEK SQL KOMUTLARI**

const TABLE\_SINEMA = 'so\_sinema';

const TABLE\_SALON = 'so\_salon';

const TABLE\_FILM = 'so\_film';

const TABLE\_GOSTERIM = 'so\_gosterim';

const TABLE\_BILET = 'so\_bilet';

const TABLE\_SATIS = 'so\_satis';

const TABLE\_SEANS = 'so\_seans';

const TABLE\_KULLANICI = 'so\_kullanici';

const TABLE\_OYUNCU = 'so\_oyuncu';

const TABLE\_YONETMEN = 'so\_yonetmen';

const TABLE\_FILM\_OYUNCU = 'so\_film\_oyuncu';

const TABLE\_FILM\_YONETMEN = 'so\_film\_yonetmen';

const TABLE\_AYARLAR = 'so\_ayarlar';

self::TABLE\_SINEMA => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_SINEMA . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

ad VARCHAR(255) NOT NULL,

yer TEXT,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_SALON => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_SALON . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

ad VARCHAR(255) NOT NULL,

kapasite INT(20) UNSIGNED NOT NULL,

ses\_sistemi VARCHAR(255),

sinema\_id INT(20) UNSIGNED NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_FILM => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_FILM . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

ad VARCHAR(255) NOT NULL,

yil INT(20) UNSIGNED,

dil VARCHAR(255),

sure VARCHAR(255) NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_GOSTERIM => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_GOSTERIM . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

salon\_id INT(20) UNSIGNED NOT NULL,

film\_id INT(20) UNSIGNED NOT NULL,

seans\_id INT(20) UNSIGNED NOT NULL,

tarih DATE NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_BILET => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_BILET . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

gosterim\_id INT(20) UNSIGNED NOT NULL,

ucret DOUBLE(5,2) NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_SATIS => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_SATIS . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

bilet\_id INT(20) UNSIGNED NOT NULL,

koltuk\_no SMALLINT(5) UNSIGNED NOT NULL,

user\_id INT(20) UNSIGNED NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_SEANS => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_SEANS . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

zaman VARCHAR(255) NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP()',

self::TABLE\_KULLANICI => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_KULLANICI . '(

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

kadi VARCHAR(255) NOT NULL,

parola VARCHAR(255) NOT NULL,

ad VARCHAR(255) NOT NULL,

email VARCHAR(255),

tel VARCHAR(255),

last\_login VARCHAR(255),

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_OYUNCU => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_OYUNCU . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

ad VARCHAR(255) NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_YONETMEN => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_YONETMEN . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

ad VARCHAR(255) NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_FILM\_OYUNCU => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_FILM\_OYUNCU . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

film\_id INT(20) UNSIGNED NOT NULL,

oyuncu\_id INT(20) UNSIGNED NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_FILM\_YONETMEN => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_FILM\_YONETMEN . ' (

id INT(20) UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

film\_id INT(20) UNSIGNED NOT NULL,

yonetmen\_id INT(20) UNSIGNED NOT NULL,

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

self::TABLE\_AYARLAR => 'CREATE TABLE IF NOT EXISTS ' . self::TABLE\_AYARLAR . ' (

setting\_name VARCHAR(255) PRIMARY KEY,

setting\_value VARCHAR(255),

stamp\_updated timestamp default now() on update now(),

stamp\_created TIMESTAMP default CURRENT\_TIMESTAMP())',

**ER Diyagramı**

Bilet

Belirle

Salon

Sinema

Satış

Admin

Kullanıcı

Giriş

Kullanıcı Girişi

Belirle

Gösterim