

**T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
İZMİR MESLEK YÜKSEKOKULU
TEKNİK PROGRAMLAR
BİLGİSAYAR PROGRAMCILIĞI BÖLÜMÜ**

SOCIAL SCOUTING NETWORK



**HAZIRLAYAN
(2020730060 – Selçukhan ÇİRGİN)**

**Danışman Öğretim Elemanı
Sedat ÇAPAR**

İzmir-2022

**T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
İZMİR MESLEK YÜKSEKOKULU
TEKNİK PROGRAMLAR
BİLGİSAYAR PROGRAMCILIĞI BÖLÜMÜ**

SOCIAL SCOUTING NETWORK



**HAZIRLAYAN
(2020730060 – Selçukhan ÇİRGİN)**

**Danışman Öğretim Elemanı
Sedat ÇAPAR**

İzmir-2022

İÇİNDEKİLER

BÖLÜM	SAYFA
KISALTMA VE SEMBOL LİSTESİ	1
ÖZET	2
GİRİŞ	3
MALZEME VE YÖNTEM	4
ANALİZ	5
BİTİRME PROJESİ ADI	5
PROJİYİ SEÇME NEDENİ	5
HİTAP EDECEĞİ SEKTÖR	5
PROJENİN GAANT ŞEMASI	6
YAZILIM DİLLERİ VE VERİTABANI	6
SWOT ANALİZİ	7
KARŞILAŞTIĞIM SORUNLAR	7
YARDIMCI OLAN UZMANLAR	8
YARDIMCI KAYNAKLAR	8
PROJENİN AKIŞ DİYAGRAMI	9
VERİ TABANI	10
PROGRAMDA KULLANILAN FORMLAR	12
PROGRAM KODLARI	23
TARTIŞMA VE SONUÇ	134
KAYNAKLAR	135
EKLER	136

KISALTMA VE SEMBOL LİSTESİ

ÖZET

Social Scouting Network türkçe adıyla sosyal futbolcu keşif sitesi Futbol Sektörü ve futbol kitlesine hizmet vermek amacıyla yapılmış futbolcuların,liglerin,takımların özelliklerini bulunduran, özel olarak kodlanmış sosyal websitesidir.

GİRİŞ

Social Scouting Network türkçe adıyla sosyal futbolcu keşif sitesi adından da belli olduğu üzere Facebook,Instagram gibi bir sosyal site, ağıdır.

Social Scouting Network'un temel amacı, site içinde olan veritabanı dahilinde futbolcuların,sporcuların çeşitli özelliklerine ulaşmak veya listede bulunmayan sporcuları veritabanına eklenmesini sağlayarak bu bilgileri diğer kullanıcılarla buluşturmaktır.Projenin benzer sitelerden en büyük farkı ise benzer sitelerde sadece futbolcuların bilgilerini görüntüleyebiliyorken Social Scouting Network de hem futbolcuların bilgilerini görüntüleyebiliyor hemde bu futbolcular hakkındaki fikirlerimizi,izlenimlerimizi paylaşabiliyoruz

MALZEME VE YÖNTEM

Proje yapılırken 2016 yılında 1.0 sürümüyle yazılım dünyasına ayak basmış ASP.NET CORE teknolojisi veya başka bir deyişle kütüphanesi kullanılmıştır.Proje ASP.NET CORE 5.0 sürümü ile geliştirilmiştir.Veritabanı “Code First” Mantığı ile oluşturulmuştur.Tasarım yani “Frontend” kısmında Bootstrap komponentleri kullanılmış olup Kod yani “Backend” kısmında ise sadece C# kullanılmıştır.Projenin içerisine Core teknolojisinin izin verdiği bir çok Framework yüklenmiş olup sıklıkla kullanılmıştır.Proje “DRY” prensibine sonuna kadar uyan mümkün olduğunca solid’i ezmeyen bir şekilde tasarlanmıştır.Veritabanı kısmında ise Microsoft SQL Server kullanılmıştır.

ANALİZ

1.Bitirme Projesi Adı

Projenin ismi Social Scouting Netwok olarak geçiyor.Türkçe olarak baz alırsak projenin ismi Sosyal Keşif Ağı olarak çevrilmektedir.

2.Projeyi Seçme Nedeni

Projeyi seçme nedenim futbola olan ilgi ve alakamdan kaynaklı.Yıllardır sıkı bir futbol ve futbolcu izleyicisi olduğum için böyle bir proje yapmak istedim.Bu projeyi lise dönemimde masaüstü versiyonu olarak daha önce amatör bir şekilde lisedeki hocalarıma sunma imkanım olmuştu.

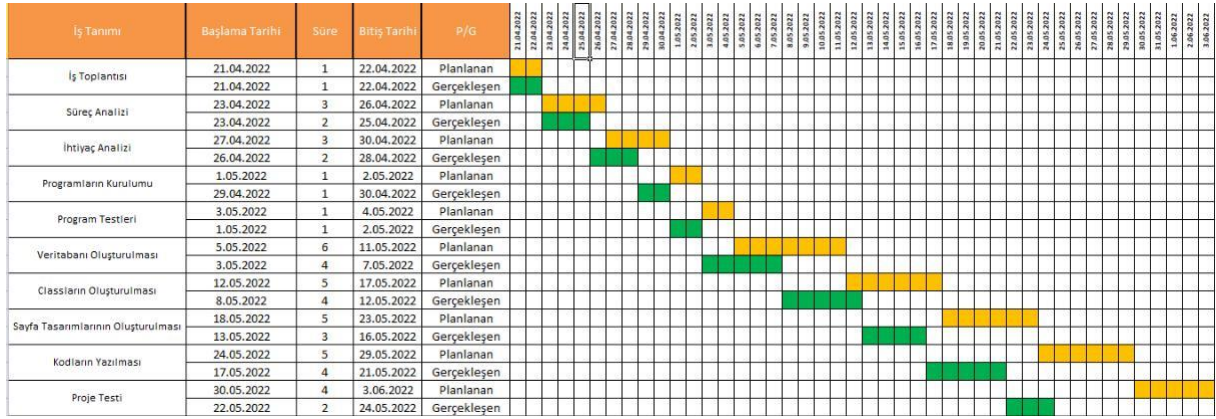
Saptadığım sorun, Diğer sitelerde sadece futbolcuların özellikleri ve bilgileri yer alıyor bu özellikleri kullanarak paylaşım yapılmasına izin verilmiyor.Sosyal ağ bu konuda yetersiz ve henüz geliştirilmemiş.

Saptadığım çözüm, Veritabanında var olan bu futbolcuların(sporcuların) özelliklerinin kullanıcılar tarafından dilediğince kullanılıp, arayüzü gelişmiş ve kaliteli bir hizmet ortaya sunmak.

3.Projenin Hitap Edeceği Sektör

Proje ilk yayınlandığı haliyle sadece futbol sektörüne hitap etmekle beraber ilerleyen zamanlarda çok yüksek ihtimalle gelişecek ve diğer spor sektörlerine de hitap edecektir.

4. Proje'nin Gantt Şeması



Proje'nin Gantt şeması görselde mevcuttur. Ek olarak daha detaylı olarak incelenmesi için projenin Gantt Şemasını dosya içinde Excel Programının destekleyeceği şekilde göndereceğim.

Planlanan kısımlar projenin daha sağlıklı olması açısından ön görülen süreleri kapsamaktadır. Fakat diğer sınavlar dolayısıyla geciktığım için 24 nisan'a kadar gerçekleşecek bir proje analizi yaptım.

5. Projede Kullanılan Yazılım Dilleri ve Veritabanı

Projede kullandığım ana yazılım dilleri ve Frameworkler:
C#, ASP.NET, MVC, MSSQL, JAVASCRIPT, BOOTSTRAP

Bu yazılım dillerini ve veritabanını seçme nedenim liseden beri C# ve MSSQL ile program yazmamdan ve bu dile diğer dillere göre çok daha iyi hakim olmamdan kaynaklanıyor.

6.Projenin SWOT Analizi

Projemin;

- 1- Güçlü Yanları : Göze hitap eden tasarım,Sosyal bir proje olması,Geliştirilebilir olması
- 2- Zayıf Yönleri : Geniş veri eksikliği,Referans eksikliği
- 3- Fırsatlar : Spor sektöründe önemi hızla artan scouting,Büyük takımlara sponsorluk
- 4- Tehditler : Benzer büyük projelerin bu projeyi taklit etmesi

Rakibimin projesinden örnek vermem gerekirse (www.transfermarkt.com); Benim projeme göre çok daha geniş bir veritabanı içeriyor. Çok daha eski ve sosyal medyada daha tanınmış bir sayfa.

Fakat kesinlikle ama kesinlikle sosyal arayüzü gerçekten çok kötü hatta düşünülmemiş(yok).Ben en başta bu sosyal arayüzü Facebook ve Instagram da olduğu gibi kullanıcıya hitap edecek bir tasarım haline getirmeyi planlamaktayım.Projemin en göz alıcı ve rakiplerimden ayrıldığı kısım bu fikirde gizlenmekte.

7.Karşılaştığım Sorunlar

Projeyi henüz yazmaya başlamadığım için kesin bilgi veremeyeceğim.Ama öncesinde MVC ile hiç proje geliştirmedığım için özellikle veritabanı bağlantısı ve codefirst ile veritabanına aktardığım örnek tablo ve ilişkilendirmelerde verisyon ve kullandığım frameworkten kaynaklı bir çok hata ile karşılaştım.

Muhtemel olarak da kendi aklımdaki kod yapısı,sayfa düzeni(tasarımı) konusunda bir çok can sıkıcı hata ile karşılaşacağım.Bunları aşmak için her zaman yaptığım gibi İngilizce olarak Google dan araştırma yapmaktayım.

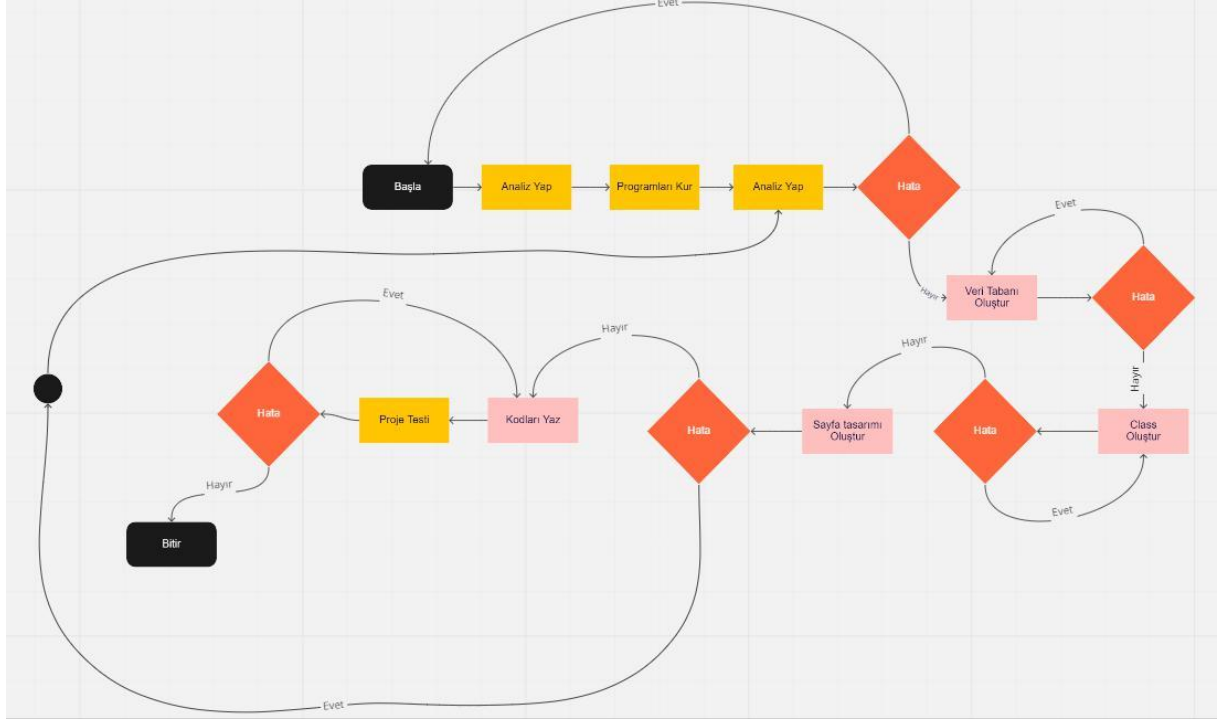
8.Bitirme Projesinde Yardımcı Olan Uzmanlar

Projemin analizini yaparken yardımcı olan bir uzmana ihtiyaç duymadım.Çünkü hem kendim futbol ile iç içe olan birisiyim hemde dışardan aldığım kaynaklar ve merakımı giderdiğim veriler olabildiğince yeterli idi.

9.Yardımcı Kaynaklar

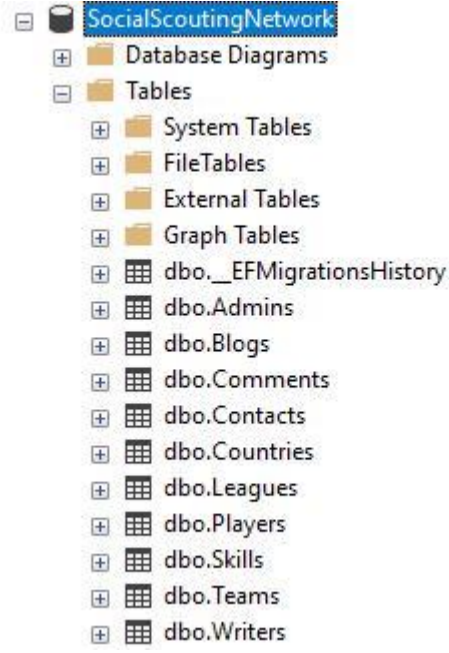
Özellikle Murat Yücedağ'ın youtube kanalındaki mvc ve core oynatma listeleri projeyi yaparken bana çok katkı sağladı.Onun dışında bazı küçük hataları giderirken stacoverflow ve microsoftun sitelerinden çokça yardım aldım.

PROJENİN AKIŞ DİYAGRAMI

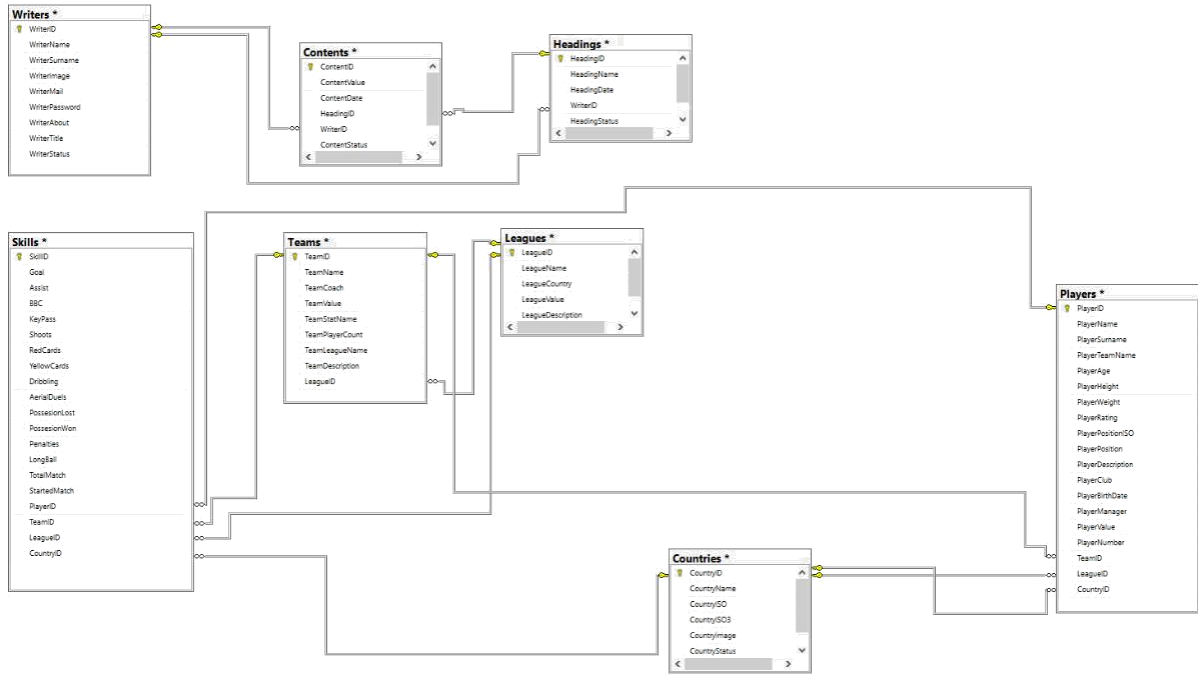


Projenin akış diyagramı resimde belirtilmiştir.

VERİ TABANI



Veritabanı adı ve tablo isimleri resimde belirtilmiştir.



Veritabanı normalizasyonu ve tablolar resimde belirtilmiştir.

PROGRAMDA KULLANILAN FORMLAR

Lig Ekleme

Lig Adı

Değeri

Takım Sayısı

Kuruluş Yılı

Açıklama

Lig Ekle








Örnek admin ekleme sayfası.

Admins CRUD

#	Admin Kullanıcı Adı	Admin Şifre	Admin Rolü	Admin Durumu	Güncelle	Sil
1	admin	admin	A	True		

Admin Ekle

Admin admins index sayfası.

#	Blog Başlığı	Blog Alanı	Küçük Resim	Büyük Resim	Blog Tarihi	Blog Durumu	Blog Yazarı	Güncelle	Sil
7	Miguel Crespo	Çok başarılı bir futbolcu gerpekten bana Appiah ı anımsatıyor		Resim Gör	17.05.2022	True	Selçukhan Çirgin	✓	✗
9	Osayi Samuel	Çok hızlı ve top hakimiyeti yüksek bir futbolcu putlarını geliştirirse önemli yerlere gelebilir		Resim Gör	17.05.2021	True	Selçukhan Çirgin	✓	✗
32	ARDA GÜLER	DASDASDASD		Resim Gör	10.06.2022	True	Selçukhan Çirgin	✓	✗
33	ARDA GÜLER	asdasdasdasdasd		Resim Gör	11.06.2022	True	Selçukhan Çirgin	✓	✗
36	ARDA GÜLER	Muazzamm		Resim Gör	11.06.2022	True	alex de souza	✓	✗
40	ARDA GÜLER	Muazzam topçu		Resim Gör	12.06.2022	True	İsmail Dıgmez	✓	✗
42	Miguel Crespo	Muazzam topçu		Resim Gör	10.06.2023	True	İsmail Dıgmez	✓	✗

Admin blog index sayfası.

#	Lig Adı	Lig Değeri	Lig Takım Sayısı	Lig Kuruluş Yılı	Lig Açıklama	Güncelle	Sil
1	Süper Lig	500	20	1959	Türkiye takımlarının oynadığı en üst seviye lig.	✓	✗
3	Premier Lig	5000000	20	1992	İngiltere takımlarının oynadığı en üst seviye lig.	✓	✗
4	La Liga	4900000	20	1929	İspanya takımlarının oynadığı en üst seviye lig.	✓	✗
5	Bundesliga	2500000	18	1962	Almanya takımlarının oynadığı en üst seviye lig.	✓	✗
6	Serie-A	3500000	20	1898	İtalya takımlarının oynadığı en üst seviye lig.	✓	✗
7	Lig 1	2650000	20	1932	Fransa takımlarının oynadığı en üst seviye lig.	✓	✗

Admin lig index sayfası.

SSN

Admin Login

Admin Login index sayfası.

- Futbolcular
- Yetenekler
- Takımlar
- Ligler
- Bloglar
- Yazarlar
- Yorumlar
- Adminler
- Çıkış Yap

Players CRUD

#	Adı	Soyadı	Yaş	Boy	Kilo	Position ISO	Position	Açıklama	Kulüp	Doğum Tarihi	Menajer	Değer	Numara	Durum	Ülke	Takım	Lig	Yetenek	Güncelle	Sil
15	Lionel	Messi	35	172	70	ODS	Ofansif Ortasaha	—	—	01.06.1987	Aile	60000000	30	True	ARGENTINA	PSG	Lig 1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Ferdi	Kadioglu	22	172	72	OS	Ortasaha	—	—	01.12.2000	Aile	5000000	25	True	NETHERLANDS	Fenerbahçe	Super Lig	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Arda	Guler	18	170	69	ODS	Ofansif Ortasaha	İyi Topçu	Fenerbahçe	01.06.2005	Aile	5000000	10	True	TURKEY	Fenerbahçe	Super Lig	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26			0	0	0					01.01.0001		5200000	0	True	AFGHANISTAN	Fenerbahçe	Super Lig	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Futbolcu Ekle

Admin futbolcular index sayfası.

Futbolcular

Yetersekler

Takımlar

Ligler

Bloglar

Yazarlar

Yorumlar

Adminler

Çıkış Yap

Teams CRUD

#	Takım Adı	Teknik Direktör	Değer	Stat Adı	Futbolcu Sayısı	Açıklama	Kuruluş Yılı	Logo	Ülke	Lig	Güncelle	Sil
1	Fenerbahçe	İsmail Kartal	125000	Ülker Stadyumu	30	En çok şampiyon olan kulüp.	1907		TURKEY	Super Lig		
2	Barcelona	Xavi	1250000	Camp Nou	29	En çok şampiyon olan 2. kulüp.	1899		SPAIN	La Liga		
3	PSG	Pochettino	15000000	Parc des Princes	28	En zengin kulüp.	1970		FRANCE	Lig 1		
4	Liverpool	Jurgen Klopp	8900000	Anfield	25	Koklu bir kulüp.	1892		UNITED KINGDOM	Premier Lig		

Takım Ekle

Admin takım index sayfası.

Futbolcular

Yetersekler

Takımlar

Ligler

Bloglar

Yazarlar

Yorumlar

Adminler

Çıkış Yap

Lig Güncelleme

Lig Adı

Lig 1

Değer

2650000

Takım Sayısı

20

Kuruluş Yılı

1932

Açıklama

Fransa takımlarının oynadığı en üst seviye lig.

Lig Güncelle

Admin örnek güncelleme sayfası.

#	Kullanıcı Adı	Adı	Soyadı	Küçük Resim	Büyük Resim	Mail	Şifre	Hakkında	Özcan	Doğum Tarihi	Kayıt Tarihi	Adres	Durum	Ülke	Güncelle	Sil
5	selcukhan1907	Selçukhan	Çirgin			selcukhancirgin@gmail.com	selcukhan	asdasdasdasd	FullStack Web Developer	05.05.2022	01.01.0001	7267 sokak no:27 dairesi:1 baraj caddesi emek mahallesi	False	TURKEY		
17	selcukhan19	Selçukhan	Çirgin			selcukhancirgin@gmail.com	admmadmm			22.06.2022	11.06.2022		True	TIMOR-LESTE		
18	alexdesouza10	Alex de	souza			alexdesouza10@gmail.com	alexdesouza			17.06.2022	11.06.2022		True	BAHGLADESH		
24	ismahdusmezz	Ismael	Düğmezz			ismahdusmezz@gmail.com	ismail23123			16.06.2022	12.06.2022		True	CAMEROON		
25	juguli	aaaa	aaa			asdasdasdasdasdasdasd	asdasdasdasd	asdasdasdasdasdasd	asdasdasd	02.06.6000	12.06.2022	sdffsdgfs	True	CROATIA		

Admin yazar index sayfası.

#	Futbolcu Adı	Gol	Asist	BBC	Aanahtar Pas	İsabetli Şut	İsabetli Şut	Kırmızı Kart	Sarı Kart	Başarısız Dripling	Başarılı Dripling	Kazanılan İkili Mücadele	Kayıbedilen İkili Mücadele	Pozisyon Kaybı	Pozisyon Kazanma	Penaltı	Uzun Top	Yedek Maç	İlk 11	Reyting	MOTM	Güncelle	Sil
1	a	9	13	36	102	87	56	0	4	14	52	54	75	21	45	6	45	5	25	8	16		
2	b	3	6	25	35	16	12	0	8	12	32	56	21	12	32	0	54	10	35	8	6		
3	c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Admin yetenek index sayfası.

Futbolcular

Yetersekler

Takımlar

Ligler

Bloglar

Yazarlar

Yorumlar

Adminler

Çıkış Yap

Comments CRUD

#	Yorum Kullanıcı Adı	Yorum Başlık	Yorum Alanı	Yorum Tarihi	Durum	Gönderi Adı	Güncelle	Sil
3	Alex	Güzel	Gerçekten çok güzel analiz yapmışsın	17.05.2022	True	Miguel Crespo		
7	asd	asds	sdad	02.06.2022	True	Miguel Crespo		

Yorum Ekle

Admin yorum index sayfası.

ScoutingNetwork

Ara

Anasayfa

Futbolcular

Takımlar

Ligler

Profilim

S.S.S

Destek

Ayarlar

Çıkış

Adres
Karıyaka/ZMİR
ScoutingSoftware, 35000

Bizi Ara
+90 539 583 75 17
...

E-mail adresimiz
info@scoutingnetwork.com
seicukhancorgm@gmail.com

Hizmet Aralıklarımız
Pazartesi - Cuma
10.00 - 19.00

Adın

Email Adresin

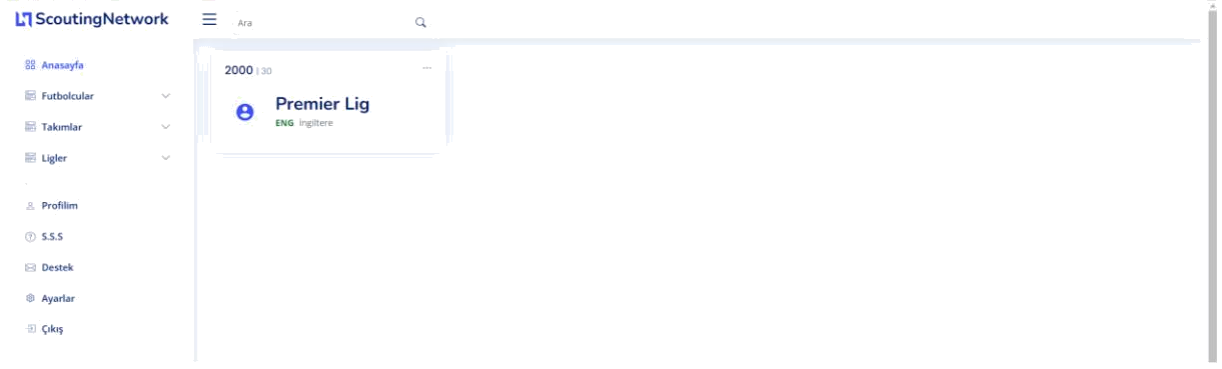
Konu

Mesaj

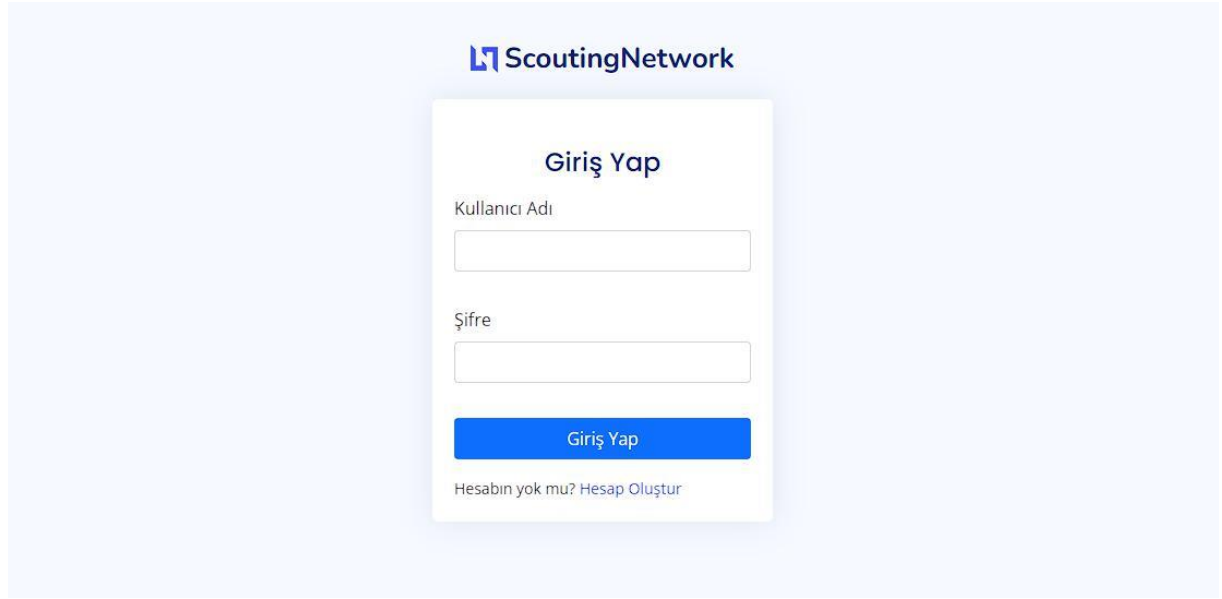
Gönder

Yazar destek sayfası.

18



Yazar ligler sayfası.



Yazar login sayfası.

ScoutingNetwork

Ara

Anasayfa
Futbolcular
Takımlar
Ligler
Profilim
S.S.S
Destek
Ayarlar
Çıkış

En Pahalı Futbolcular

En Çok Asist Yapanlar

Selçukhan Çiğın 17.05.2022

Miguel Crespo
Çok savaşçı bir futbolcu gerçekten bana Appiah ı anımsatıyor
[Devamını Oku ...](#)

Selçukhan Çiğın 17.05.2021

Osayi Samuel
Çok hızlı ve top hakimiyeti yüksek bir futbolcu şutlarını geliştirirse önemli yerlere gelebilir
[Devamını Oku ...](#)

Selçukhan Çiğın 10.06.2022

ARDA GÜLER
DASDASDASD
[Devamını Oku ...](#)

Süper Lig Standings

#	Team	P	W	D	L	PTS
1	Trabzonspor	38	23	12	3	81
2	Fenerbahçe	38	21	10	7	73
3	Konyaspor	38	20	8	10	68
4	İstanbul Başakşehir	38	19	8	11	65
5	Alanyaspor	38	19	7	12	64
6	Beşiktaş	38	15	14	9	59
7	Antalyaspor	38	16	11	11	59
8	Fatih Karagömrük	38	16	9	13	57
9	Adana Demirspor	38	15	10	13	55
10	Sivasspor	38	14	12	12	54

En Değerli 3 Futbolcu

30 | PSG

Lionel Messi
OOS ARGENTINA

0 | Fenerbahçe

AFGHANISTAN

25 | Fenerbahçe

Ferdi Kadioğlu

Yazar anasayfa.

ScoutingNetwork

Ara

Anasayfa
Futbolcular
Takımlar
Ligler
Profilim
S.S.S
Destek
Ayarlar
Çıkış

30 | PSG

Lionel Messi
OOS ARGENTINA

25 | Fenerbahçe

Ferdi Kadioğlu
OS NETHERLANDS

10 | Fenerbahçe

Arda Güler
OOS TURKEY

0 | Fenerbahçe

AFGHANISTAN

Yazar futbolcular sayfası.

ScoutingNetwork

Ara

Anasayfa

Futbolcular

Takımlar

Ligler

Profilim


S.S.S

Destek

Ayarlar

Çıkış


Gönderi Oluştur



selcukhan1907
FullStack Web Developer

Selçukhan Çirgin


17.05.2022



Miguel Crespo
Çok savaşçı bir futbolcu gerçekten bana Appiah'ı anımsatıyor
[Devamını Oku ...](#)

Selçukhan Çirgin


17.05.2021



Osayl Samuel
Çok hızlı ve top hakimiyeti yüksek bir futbolcu şutlarını geliştirirse önemli yerlere gelebilir
[Devamını Oku ...](#)

Selçukhan Çirgin

10.06.2022



ARDA GÜLER

Super Lig

Standings

VIEW MORE

#	Team	P	W	D	L	PTS
1	Trabzonspor	38	23	12	3	81
2	Fenerbahçe	38	21	10	7	73
3	Konyaspor	38	20	8	10	68
4	İstanbul Başakşehir	38	19	8	11	65
5	Alanyaspor	38	19	7	12	64
6	Beyköş	38	15	14	9	59
7	Antalyaspor	38	16	11	11	59
8	Fatih Karagömrük	38	16	9	13	57
9	Adana Demirspor	38	15	10	13	55
10	Sivasspor	38	14	12	12	54

En Değerli 3 Futbolcu

30 | PSG

Lionel Messi

ARGENTINA

0 | Fenerbahçe

AFGHANISTAN

Yazar profil sayfası.

ScoutingNetwork

Hesap Oluştur

Ad

Soyad

Kullanıcı Adı

Email

Şifre

Profil Resmi

Dosya Seç

Dosya seçilmedi

Doğum Tarihi

Ülke

gg.aa.yyyy

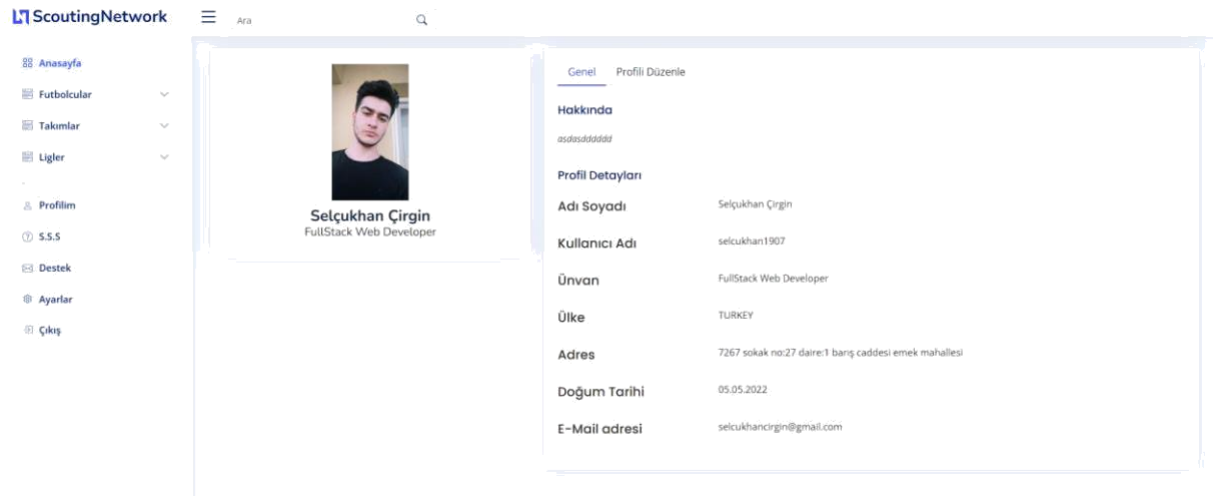
AFGHANISTAN

Hesap Oluştur

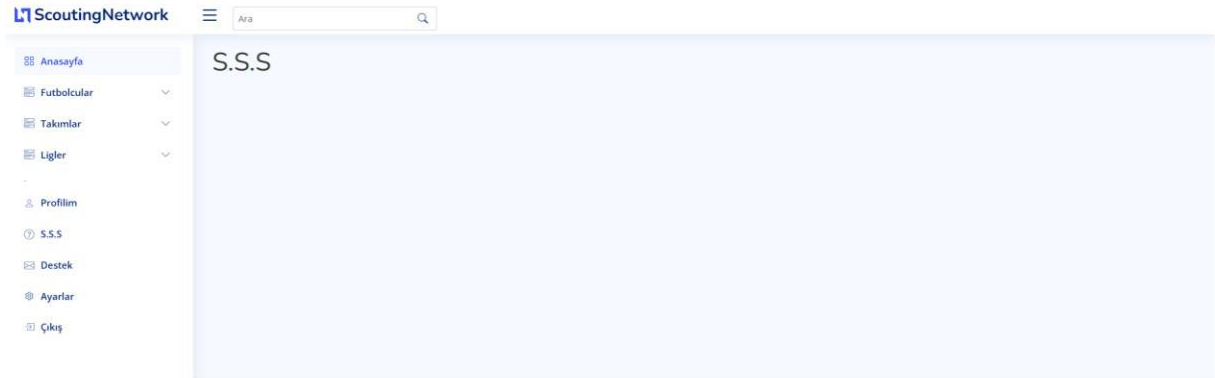
Zaten hesabın var mı? Giriş yap

Yazar kayıt sayfası.

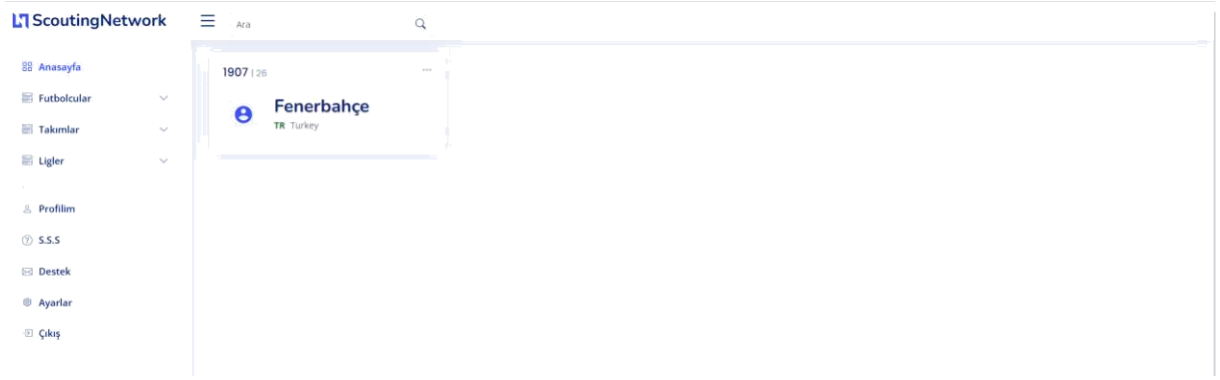
21



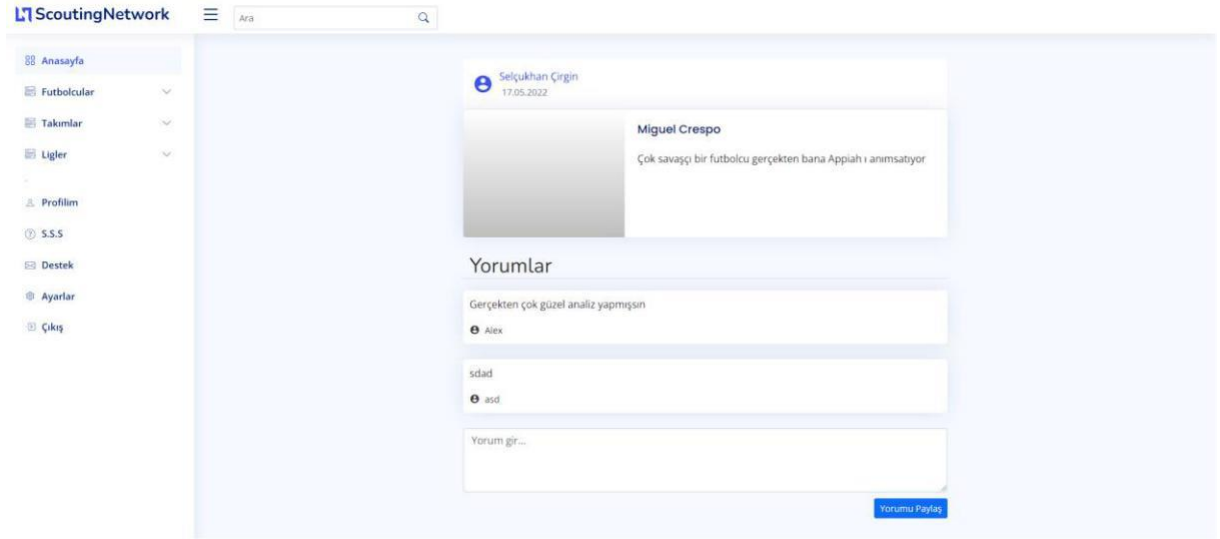
Yazar ayarlar sayfası.



Yazar S.S.S sayfası.



Yazar takımlar sayfası.



Yazar yorumlar sayfası.

PROGRAM KODLARI

I. Business Layer Abstract

a. IServices :

J.

```
public interface IAdminService
{
    void AddAdmin(Admin admin);
    void DeleteAdmin(Admin admin);
    void UpdateAdmin(Admin admin);
    List<Admin> GetList();
    Admin GetByID(int id);
}

public interface IBlogService
{
    void AddBlog(Blog blog);
    void DeleteBlog(Blog blog);
    void UpdateBlog(Blog blog);
    List<Blog> GetList();
    Blog GetByID(int id);
    List<Blog> GetListWithAllAttribution();
    List<Blog> GetBlogListByWriter(int id);
}

public interface ICommentService
{
    void AddComment(Comment comment);
    //void DeleteComment(Comment comment);
    //void UpdateComment(Comment comment);
    List<Comment> GetListWithAllAttribution();
    List<Comment> GetList(int id);
    //Comment GetByID(int id);
    //List<Comment> GetListWithAllAttribution();
}

public interface IContactService
{
    void AddContact(Contactt contact);
}

public interface ICountryService
{
    List<Country> GetListCountry();
}

public interface ILeagueService
{
    void AddLeague(League league);
    void DeleteLeague(League league);
    void UpdateLeague(League league);
    List<League> GetList();
    League GetByID(int id);
    List<League> GetListWithAllAttribution();
}

public interface IPlayerService
{
    void AddPlayer(Player player);
}
```

```
void DeletePlayer(Player player);  
void UpdatePlayer(Player player);  
List<Player> GetList();  
Player GeyByID(int id);  
List<Player> GetListWithAllAttribution();  
}
```

```

public interface ISkillService
{
    void AddSkill(Skill skill);
    void DeleteSkill(Skill skill);
    void UpdateSkill(Skill skill);
    List<Skill> GetList();
    Skill GetByID(int id);
    List<Skill> GetListWithAllAttribution();
}

public interface ITeamService
{
    void AddTeam(Team team);
    void DeleteTeam(Team team);
    void UpdateTeam(Team team);
    List<Team> GetList();
    Team GetByID(int id);
    List<Team> GetListWithAllAttribution();
}

public interface IWriterService
{
    void AddWriter(Writer writer);
    void DeleteWriter(Writer writer);
    void UpdateWriter(Writer writer);
    List<Writer> GetList();
    Writer GetByID(int id);
}

```

I. Business Layer Concrete

a. Managers:

```
public class AdminManager : IAdminService
{
    IAdminDal _adminDal;

    public AdminManager(IAdminDal adminDal)
    {
        _adminDal = adminDal;
    }

    public void AddAdmin(Admin admin)
    {
        _adminDal.Insert(admin);
    }

    public void DeleteAdmin(Admin admin)
    {
        _adminDal.Delete(admin);
    }

    public Admin GetByID(int id)
    {
        return _adminDal.GetByID(id);
    }

    public List<Admin> GetList()
    {
        return _adminDal.GetListAll();
    }

    public void UpdateAdmin(Admin admin)
    {
        _adminDal.Update(admin);
    }
}

public class BlogManager : IBlogService
{
    IBlogDal _blogDal;
    public BlogManager(IBlogDal blogDal)
    {
        _blogDal = blogDal;
    }

    public void AddBlog(Blog blog)
    {
        _blogDal.Insert(blog);
    }

    public void DeleteBlog(Blog blog)
    {
        _blogDal.Delete(blog);
    }

    public List<Blog> GetList()
    {
        return _blogDal.GetListAll();
    }

    public List<Blog> GetListWithAllAttribution()
```

```

{
    return _blogDal.GetListWithAllAttribution();
}

public Blog GetByID(int id)
{
    return _blogDal.GetByID(id);
}

public List<Blog> GetListWithAllAttribution2(int id)
{
    return _blogDal.GetListWithAllAttribution2(x => x.BlogID == id);
}
public List<Blog> GB(int id)
{
    return _blogDal.GetListAll(x => x.BlogID == id);
}
public void UpdateBlog(Blog blog)
{
    _blogDal.Update(blog);
}

public List<Blog> GetBlogListByWriter(int id)
{
    return _blogDal.GetListAll(x => x.WriterID == id);
}
    }

```

```

public class CommentManager : ICommentService
{
    ICommentDal _commentDal;

    public CommentManager(ICommentDal commentDal)
    {
        _commentDal = commentDal;
    }

    public void AddComment(Comment comment)
    {
        throw new NotImplementedException();
    }

    public List<Comment> GetList(int id)
    {
        return _commentDal.GetListAll(x => x.BlogID == id);
    }

    public List<Comment> GetListWithAllAttribution()
    {
        return _commentDal.GetListWithAllAttribution();
    }
}

public class ContactManager : IContactService
{
    IContactDal _contactDal;

    public ContactManager(IContactDal contactDal)
    {
        _contactDal = contactDal;
    }

    public void AddContact(Contactt contact)
    {
        _contactDal.Insert(contact);
    }
}

public class CountryManager : ICountryService
{
    ICountryDal _countryDal;

    public CountryManager(ICountryDal countryDal)
    {
        _countryDal = countryDal;
    }

    public List<Country> GetListCountry()
    {
        return _countryDal.GetListAll();
    }
}

```

```

public class LeagueManager : ILeagueService
{
    ILeagueDal _leagueDal;

    public LeagueManager(ILeagueDal leagueDal)
    {
        _leagueDal = leagueDal;
    }

    public void AddLeague(League league)
    {
        _leagueDal.Insert(league);
    }

    public void DeleteLeague(League league)
    {
        _leagueDal.Delete(league);
    }

    public League GetByID(int id)
    {
        return _leagueDal.GetByID(id);
    }

    public List<League> GetList()
    {
        return _leagueDal.GetListAll();
    }

    public List<League> GetListWithAllAttribution()
    {
        throw new NotImplementedException();
    }

    public void UpdateLeague(League league)
    {
        _leagueDal.Update(league);
    }
}

```



```

public class PlayerManager : IPlayerService
{
    IPlayerDal _playerDal;

    public PlayerManager(IPlayerDal playerDal)
    {
        _playerDal = playerDal;
    }

    public void AddPlayer(Player player)
    {
        _playerDal.Insert(player);
    }

    public void DeletePlayer(Player player)
    {
        _playerDal.Delete(player);
    }

    public List<Player> GetList()
    {
        return _playerDal.GetListAll();
    }

    public List<Player> GetListWithAllAttribution()
    {
        return _playerDal.GetListWithAllAttribution();
    }

    public List<Player> GetListWithAllAttribution2(int id)
    {
        return _playerDal.GetListWithAllAttribution2(x => x.PlayerID == id);
    }

    public Player GeyByID(int id)
    {
        return _playerDal.GetByID(id);
    }

    public void UpdatePlayer(Player player)
    {
        _playerDal.Update(player);
    }
}

```

```

public class SkillManager : ISkillService
{
    ISkillDal _skillDal;

    public SkillManager(ISkillDal skillDal)
    {
        _skillDal = skillDal;
    }

    public void AddSkill(Skill skill)
    {
        _skillDal.Insert(skill);
    }

    public void DeleteSkill(Skill skill)
    {
        _skillDal.Delete(skill);
    }

    public Skill GetByID(int id)
    {
        return _skillDal.GetByID(id);
    }

    public List<Skill> GetList()
    {
        return _skillDal.GetListAll();
    }

    public List<Skill> GetListWithAllAttribution()
    {
        throw new NotImplementedException();
    }

    public void UpdateSkill(Skill skill)
    {
        _skillDal.Update(skill);
    }
}

```

```

public class TeamManager : ITeamService
{
    ITeamDal _teamDal;

    public TeamManager(ITeamDal teamDal)
    {
        _teamDal = teamDal;
    }

    public void AddTeam(Team team)
    {
        _teamDal.Insert(team);
    }

    public void DeleteTeam(Team team)
    {
        _teamDal.Delete(team);
    }

    public Team GetByID(int id)
    {
        return _teamDal.GetByID(id);
    }

    public List<Team> GetList()
    {
        return _teamDal.GetListAll();
    }

    public List<Team> GetListWithAllAttribution()
    {
        return _teamDal.GetListWithAllAttribution();
    }

    public void UpdateTeam(Team team)
    {
        _teamDal.Update(team);
    }
}

```

```

public class WriterManager : IWriterService
{
    IWriterDal _writerDal;

    public WriterManager(IWriterDal writerDal)
    {
        _writerDal = writerDal;
    }

    public void AddWriter(Writer writer)
    {
        _writerDal.Insert(writer);
    }

    public void DeleteWriter(Writer writer)
    {
        _writerDal.Delete(writer);
    }

    public Writer GetByID(int id)
    {
        return _writerDal.GetByID(id);
    }
    public List<Writer> GetListWithAllAttribution()
    {
        return _writerDal.GetListWithAllAttribution();
    }
    public List<Writer> GetList()
    {
        return _writerDal.GetListAll();
    }
    public List<Writer> GetListWithAllAttribution2(int id)
    {
        return _writerDal.GetListWithAllAttribution2(x => x.WriterID == id);
    }

    public void UpdateWriter(Writer writer)
    {
        _writerDal.Update(writer);
    }
}

```

İİ. Business Layer

ValidationRules a. AllRules

```
public class RegisterValidator : AbstractValidator<Writer>
{
    public RegisterValidator()
    {
        RuleFor(x => x.WriterName).NotEmpty().WithMessage("Yazar adı boş
geçilemez!");
        RuleFor(x => x.WriterName).MaxLength(25).WithMessage("Yazar
adı maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterName).MinLength(2).WithMessage("Yazar adı
minimum 2 karakter olmalı!");

        RuleFor(x => x.WriterSurname).NotEmpty().WithMessage("Yazar soyadı
boş geçilemez!");
        RuleFor(x => x.WriterSurname).MaxLength(25).WithMessage("Yazar soyadı
maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterSurname).MinLength(2).WithMessage("Yazar soyadı
minimum 2 karakter olmalı!");

        RuleFor(x => x.WriterUserName).NotEmpty().WithMessage("Yazar kullanıcı
adı boş geçilemez!");
        RuleFor(x => x.WriterUserName).MaxLength(25).WithMessage("Yazar
kullanıcı adı maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterUserName).MinLength(6).WithMessage("Yazar
kullanıcı adı minimum 6 karakter olmalı!");

        RuleFor(x => x.WriterSurname).NotEmpty().WithMessage("Yazar soyadı
boş geçilemez!");
        RuleFor(x => x.WriterSurname).MaxLength(25).WithMessage("Yazar soyadı
maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterSurname).MinLength(2).WithMessage("Yazar soyadı
minimum 2 karakter olmalı!");

        RuleFor(x => x.WriterMail).NotEmpty().WithMessage("Yazar maili
boş geçilemez!");
        RuleFor(x => x.WriterMail).MaxLength(100).WithMessage("Yazar
maili maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterMail).MinLength(15).WithMessage("Yazar
maili minimum 10 karakter olmalı!");
        RuleFor(x => x.WriterMail).EmailAddress().WithMessage("Email formatı
hatalı!");

        RuleFor(x => x.WriterPassword).NotEmpty().WithMessage("Yazar şifresi boş
geçilemez!");
        RuleFor(x => x.WriterPassword).MaxLength(25).WithMessage("Yazar
şifresi maksimum 25 karakter olmalı!");
        RuleFor(x => x.WriterPassword).MinLength(8).WithMessage("Yazar
şifresi minimum 8 karakter olmalı!");

        //RuleFor(x => x.WriterImage).NotEmpty().WithMessage("Yazar Resmi
boş geçilemez!");
    }
}
```

```

public class LoginValidator : AbstractValidator<Writer>
{
    private bool UniqueName(Writer writer, string name)
    {
        if (name == null || name == "")
        {
            return true;
        }

        Context _db = new Context();
        var dbCategory = _db.Writers
            .Where(x => x.WriterUserName.ToLower() ==
name.ToLower())
            .FirstOrDefault();

        if (dbCategory == null)
        {
            return false;
        }
        else
            return true;
    }
    private bool UniquePassword(Writer writer, string password)
    {
        if (password == null || password == "")
        {
            return true;
        }

        Context _db = new Context();
        var dbCategory = _db.Writers
            .Where(x => x.WriterPassword.ToLower() ==
password.ToLower())
            .FirstOrDefault();

        if (dbCategory == null)
        {
            return false;
        }
        else
            return true;
    }

    public LoginValidator()
    {
        RuleFor(x => x.WriterUserName).NotEmpty().WithMessage("Yazar kullanıcı adı
boş geçilemez!").Must(UniqueName).WithMessage("Kullanıcı adı hatalı!");
        RuleFor(x => x.WriterPassword).NotEmpty().WithMessage("Yazar şifresi boş
geçilemez!").Must(UniquePassword).WithMessage("Kullanıcı şifresi hatalı!");
    }
}

```

```

public class AdminLoginValidator: AbstractValidator<Admin>
{
    private bool UniqueName(Admin writer, string name)
    {
        if (name == null || name == "")
        {
            return true;
        }

        Context _db = new Context();
        var dbCategory = _db.Admins
            .Where(x => x.AdminUserName.ToLower() ==
name.ToLower())
            .FirstOrDefault();

        if (dbCategory == null)
        {
            return false;
        }
        else
            return true;
    }
    private bool UniquePassword(Admin writer, string password)
    {
        if (password == null || password == "")
        {
            return true;
        }

        Context _db = new Context();
        var dbCategory = _db.Admins
            .Where(x => x.AdminPassword.ToLower() ==
password.ToLower())
            .FirstOrDefault();

        if (dbCategory == null)
        {
            return false;
        }
        else
            return true;
    }
}
public AdminLoginValidator()
{
    RuleFor(x => x.AdminUserName).NotEmpty().WithMessage("Admin kullanıcı adı
boş geçilemez!").Must(UniqueName).WithMessage("Kullanıcı kullanıcı adı hatalı!");
    RuleFor(x => x.AdminPassword).NotEmpty().WithMessage("Admin şifresi boş
geçilemez!").Must(UniquePassword).WithMessage("Admin şifresi hatalı!");
}
}

```

III. Data Acces Layer Abstract

```
a. IDals :
    public interface IAdminDal: IGenericDal<Admin>
    {
    }
    public interface IBlogDal : IGenericDal<Blog>
    {
        List<Blog> GetListWithAllAttribution();
        List<Blog> GetListWithAllAttribution2(Expression<Func<Blog, bool>> filter);
    }
public interface ICommentDal : IGenericDal<Comment>
{
    List<Comment> GetListWithAllAttribution();
}

public interface IContactDal : IGenericDal<Contactt>
{
}

public interface ICountryDal : IGenericDal<Country>
{
}

public interface IGenericDal<T> where T : class
{
    void Insert(T t);
    void Delete(T t);
    void Update(T t);
    List<T> GetListAll();
    T GetByID(int id);
    List<T> GetListAll(Expression<Func<T, bool>>filter);
}

public interface ILeagueDal : IGenericDal<League>
{
}

public interface IPlayerDal : IGenericDal<Player>
{
    List<Player> GetListWithAllAttribution();
    List<Player> GetListWithAllAttribution2(Expression<Func<Player, bool>>
filter);
}

public interface ISkillDal : IGenericDal<Skill>
{
}

public interface ITeamDal : IGenericDal<Team>
{
    List<Team> GetListWithAllAttribution();
}

public interface IWriterDal : IGenericDal<Writer>
{
    List<Writer> GetListWithAllAttribution();
    List<Writer> GetListWithAllAttribution2(Expression<Func<Writer, bool>>
filter);
}
```


I. Data Acces Layer Concrete
a. Context

```
public class Context : DbContext
{
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer("Server=DESKTOP-
K2JUCK9;Database=SocialScoutingNetwork;Integrated Security=True;");
    }

    public DbSet<Blog> Blogs { get; set; }
    public DbSet<Comment> Comments { get; set; }
    public DbSet<Contactt> Contacts { get; set; }
    } public DbSet<Country> Countries { get;
set; } public DbSet<League> Leagues { get;
set; } public DbSet<Player> Players { get;
set; } public DbSet<Skill> Skills { get;
set; } public DbSet<Team> Teams { get; set;
} public DbSet<Writer> Writers { get; set; }
public DbSet<Admin> Admins { get; set; }
}
```

I. Data Acces Layer EntityFramework

a. Repositories

```
public class EfAdminRepository:GenericRepository<Admin>, IAdminDal
{
}

public class EfBlogRepository : GenericRepository<Blog>, IBlogDal
{
    public List<Blog> GetListWithAllAttribution()
    {
        using (var c = new Context())
        {
            return c.Blogs.Include(x => x.Writer).ToList();
        }
    }
    public List<Blog> GetListWithAllAttribution2(Expression<Func<Blog, bool>>
filter)
    {
        using (var k = new Context())
        {
            return k.Set<Blog>().Include(x => x.Writer).Where(filter).ToList();
        }
    }
}

public class EfCommentRepository : GenericRepository<Comment>, ICommentDal
{
    public List<Comment> GetListWithAllAttribution()
    {
        using (var c = new Context())
        {
            return c.Comments.Include(x => x.Blog).ToList();
        }
    }
}

public class EfContactRepository : GenericRepository<Contactt>, IContactDal
{
}

public class EfCountryRepository : GenericRepository<Country>, ICountryDal
{
}

public class EfLeagueRepository : GenericRepository<League>, ILeagueDal
{
}

public class EfPlayerRepository : GenericRepository<Player>, IPlayerDal
{
    public List<Player> GetListWithAllAttribution()
    {
        using (var c = new Context())
        {
            return c.Players.Include(x => x.Country).Include(x
=> x.League).Include(x => x.Skill).Include(x => x.Team).ToList();
        }
    }
    public List<Player> GetListWithAllAttribution2(Expression<Func<Player, bool>>
filter)
    {
        using (var k = new Context())
        {
            return k.Set<Player>().Include(x => x.Country).Include(x =>
x.League).Include(x => x.Skill).Include(x => x.Team).Where(filter).ToList();
            //return k.Blogs.Include(x => x.Writer).ToList();
        }
    }
}
```

```

public class EfSkillRepository : GenericRepository<Skill>, ISkillDal
{
}
public class EfTeamRepository : GenericRepository<Team>, ITeamDal
{
    public List<Team> GetListWithAllAttribution()
    {
        using (var c = new Context())
        {
            return c.Teams.Include(x => x.Country).Include(x
=> x.League).ToList();
        }
    }
}
public class EfWriterRepository : GenericRepository<Writer>, IWriterDal
{
    public List<Writer> GetListWithAllAttribution()
    {
        using (var c = new Context())
        {
            return c.Writers.Include(x => x.Country).ToList();
        }
    }

    public List<Writer> GetListWithAllAttribution2(Expression<Func<Writer, bool>>
filter)
    {
        using (var k = new Context())
        {
            return k.Set<Writer>().Include(x => x.Country).Where(filter).ToList();
            //return k.Blogs.Include(x => x.Writer).ToList();
        }
    }
}
}

```

II. Data Acces Layer Repositories

a. GenericRepository

```
public class GenericRepository<T> : IGenericDal<T> where T : class
{
    public void Delete(T t)
    {
        using var c = new Context();
        c.Remove(t);
        c.SaveChanges();
    }

    public T GetByID(int id)
    {
        using var c = new Context();
        return c.Set<T>().Find(id);
    }

    public List<T> GetListAll()
    {
        using var c = new Context();
        return c.Set<T>().ToList();
    }

    public void Insert(T t)
    {
        using var c = new Context();
        c.Add(t);
        c.SaveChanges();
    }

    public List<T> GetListAll(Expression<Func<T, bool>> filter)
    {
        using var c = new Context();
        return c.Set<T>().Where(filter).ToList();
    }

    public void Update(T t)
    {
        using var c = new Context();
        c.Update(t);
        c.SaveChanges();
    }
}
```

EntityLayer Concrete

```
public class Admin
{
    [Key]
    public int AdminID { get; set; }
    public string AdminUserName { get; set; }
    public string AdminPassword { get; set; }
    public string AdminRole { get; set; }
    public bool AdminStatus { get; set; }
}

public class Blog
{
    [Key]
    public int BlogID { get; set; }
    public string BlogTitle { get; set; }
    public string BlogContent { get; set; }
    public string BlogThumbnailImage { get; set; }
    public string BlogImage { get; set; }
    public DateTime BlogCreateDate { get; set; }
    public bool BlogStatus { get; set; }
    public int WriterID { get; set; }
    public Writer Writer { get; set; }
    public List<Comment> Comments { get; set; }
}

public class Comment
{
    [Key]
    public int CommentID { get; set; }
    public string CommentUserName { get; set; }
    public string CommentTitle { get; set; }
    public string CommentContent { get; set; }
    public DateTime CommentDate { get; set; }
    public bool CommentStatus { get; set; }
    public int BlogID { get; set; }
    public Blog Blog { get; set; }
}

public class Contactt
{
    [Key]
    public int ContactID { get; set; }
    public string ContactUserName { get; set; }
    public string ContactMail { get; set; }
    public string ContactSubject { get; set; }
    public string ContactMessage { get; set; }
    public DateTime ContactDate { get; set; }
    public bool ContactStatus { get; set; }
}

public class Country
{
    [Key]
    public int CountryID { get; set; }
```

```

        public string CountryISO { get; set; }
        public string CountryName { get; set; }
        public string CountryNName { get; set; }
        public string CountryISO3 { get; set; }
        public int CountryNumCode { get; set; }
        public int CountryPhoneCode { get; set; }
        public List<Writer> Writers { get; set; }
        public List<Player> Players { get; set; }
        public List<Team> Teams { get; set; }
    }
}

public class League
{
    [Key]
    public int LeagueID { get; set; }
    public string LeagueName { get; set; }
    public int LeagueValue { get; set; }
    public int LeagueTeamCount { get; set; }
    public int LeagueFoundationYear { get; set; }
    public string LeagueDescription { get; set; }
    public List<Player> Players { get; set; }
    public List<Team> Teams { get; set; }
}

public class Player
{
    [Key]
    public int PlayerID { get; set; }
    public string PlayerName { get; set; }
    public string PlayerSurname { get; set; }
    public int PlayerAge { get; set; }
    public int PlayerHeight { get; set; }
    public int PlayerWeight { get; set; }
    public string PlayerPositionISO { get; set; }
    public string PlayerPosition { get; set; }
    public string PlayerDescription { get; set; }
    public string PlayerClub { get; set; }
    public DateTime PlayerBirthDate { get; set; }
    public string PlayerManager { get; set; }
    public int PlayerValue { get; set; }
    public int PlayerNumber { get; set; }
    public bool PlayerStatus { get; set; }
    public int CountryID { get; set; }
    public Country Country { get; set; }
    public int TeamID { get; set; }
    public Team Team { get; set; }
    public int LeagueID { get; set; }
    public League League { get; set; }
    public int SkillID { get; set; }
    public Skill Skill { get; set; }
}

}

public class Skill
{
    [Key]
    public int SkillID { get; set; }

```

```

        public string SkillPlayerName { get; set; }
        public int Goal { get; set; }
        public int Assist { get; set; }
        public int BBC { get; set; }
        public int KeyPass { get; set; }
        public int ShootsOffTarget { get; set; }
        public int ShootsOnTarget { get; set; }
        public int RedCards { get; set; }
        public int YellowCards { get; set; }
        public int UnsuccessfulDribbling { get; set; }
        public int SuccesfulDribbling { get; set; }
        public int AerialDuelsWon { get; set; }
        public int AerialDuelsLost { get; set; }
        public int PossesionLost { get; set; }
        public int PossesionWon { get; set; }
        public int Penalties { get; set; }
        public int LongBall { get; set; }
        public int BenchMatch { get; set; }
        public int StartedMatch { get; set; }
        public int Rating { get; set; }
        public int MOTM { get; set; }
        public List<Player> Players { get; set; }
    }
    public class Team
    {
        [Key]
        public int TeamID { get; set; }
        public string TeamName { get; set; }
        public string TeamCoach { get; set; }
        public int TeamValue { get; set; }
        public string TeamStatName { get; set; }
        public int TeamPlayerCount { get; set; }
        public string TeamDescription { get; set; }
        public int TeamFoundationYear { get; set; }
        public string TeamImage { get; set; }
        public int CountryID { get; set; }
        public Country Country { get; set; }
        public int LeagueID { get; set; }
        public League League { get; set; }
        public List<Player> Players { get; set; }
    }
    public class Writer
    {
        [Key]
        public int WriterID { get; set; }
        public string WriterUserName { get; set; }
        public string WriterName { get; set; }
        public string WriterSurname { get; set; }
        public string WriterImage { get; set; }
        public string WriterMail { get; set; }
        public string WriterPassword { get; set; }
        public string WriterAbout { get; set; }
        public string WriterTitle { get; set; }
        public DateTime WriterBirthDate { get; set; }
        public DateTime WriterCreateDate { get; set; }
        public string WriterAddress { get; set; }
        public int CountryID { get; set; }
        public Country Country { get; set; }
        public bool WriterStatus { get; set; }
        public List<Blog> Blogs { get; set; }
    }
}

```

Controllers

```
public class AdminController : Controller
{
    PlayerManager pm = new PlayerManager(new EfPlayerRepository());
    CountryManager cm = new CountryManager(new EfCountryRepository());
    TeamManager tm = new TeamManager(new EfTeamRepository());
    LeagueManager lm = new LeagueManager(new EfLeagueRepository());
    SkillManager sm = new SkillManager(new EfSkillRepository());
    BlogManager blgm = new BlogManager(new EfBlogRepository());
    WriterManager wm = new WriterManager(new EfWriterRepository());
    CommentManager cmnt = new CommentManager(new EfCommentRepository());
    AdminManager am = new AdminManager(new EfAdminRepository());
    CountryManager cou = new CountryManager(new EfCountryRepository());
    ContactManager cnt = new ContactManager(new EfContactRepository());

    [AllowAnonymous]
    public IActionResult AdminLoginIndex(Admin p)
    {
        AdminLoginValidator alv = new AdminLoginValidator();
        ValidationResult results = alv.Validate(p);

        var c = new Context();
        var datavalue = c.Admins.FirstOrDefault(x => x.AdminUserName ==
p.AdminUserName && x.AdminPassword == p.AdminPassword);

        if (results.IsValid)
        {
            if (datavalue != null)
            {
                return RedirectToAction("Index", "Admin");
            }
        }
        else
        {
            foreach (var item in results.Errors)
            {
                ModelState.AddModelError(item.PropertyName, item.ErrorMessage);
            }
        }

        return View();
    }
    public async Task<IActionResult> Logout()
    {
        await
HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
        return RedirectToAction("AdminLoginIndex", "Admin");
    }
    public IActionResult ContactIndex()
    {
        var values = cnt.GetList();
        return View(values);
    }

    public IActionResult CountryIndex()
    {
        var values = cou.GetListCountry();
        return View(values);
    }
}
```



```

public IActionResult AdminIndex()
{
    var values = am.GetList();
    return View(values);
}

public IActionResult Index()
{
    var values = pm.GetListWithAllAttribution().ToList();
    return View(values);
}
public IActionResult SkillIndex()
{
    var values = sm.GetList().ToList();
    return View(values);
}
public IActionResult TeamIndex()
{
    var values = tm.GetListWithAllAttribution().ToList();
    return View(values);
}
public IActionResult LeagueIndex()
{
    var values = lm.GetList();
    return View(values);
}

public IActionResult BlogIndex()
{
    var values = blgm.GetListWithAllAttribution().ToList();
    return View(values);
}

public IActionResult WriterIndex()
{
    var values = wm.GetListWithAllAttribution().ToList();
    return View(values);
}
public IActionResult CommentIndex()
{
    var values = cmnt.GetListWithAllAttribution().ToList();
    return View(values);
}

[HttpGet]
public IActionResult AddPlayer()
{
    List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                        select new SelectListItem
                                        {
                                            Text = x.CountryName,
                                            Value = x.CountryID.ToString()
                                        }).ToList();

    ViewBag.country = valuecountry;

    List<SelectListItem> valueteam = (from x in tm.GetList()
                                     select new SelectListItem
                                     {
                                         Text = x.TeamName,
                                         Value = x.TeamID.ToString()

```

```

        }
        ).ToList();

        ViewBag.team = valueteam;

        List<SelectListItem> valueleague = (from x in lm.GetList()
        select new SelectListItem
        {
            Text = x.LeagueName,
            Value = x.LeagueID.ToString()
        }
        ).ToList();

        ViewBag.league = valueleague;

        List<SelectListItem> valueskill = (from x in sm.GetList()
        select new SelectListItem
        {
            Text = x.SkillPlayerName,
            Value = x.SkillID.ToString()
        }
        ).ToList();

        ViewBag.skill = valueskill;

        return View();
    }

    [HttpPost]
    public IActionResult AddPlayer(Player p)
    {
        p.PlayerStatus = true;
        p.SkillID = 3;
        pm.AddPlayer(p);
        return RedirectToAction("Index", "Admin");
    }

    [HttpGet]
    public IActionResult AddSkill()
    {
        return View();
    }

    [HttpPost]
    public IActionResult AddSkill(Skill p)
    {
        sm.AddSkill(p);
        return RedirectToAction("SkillIndex", "Admin");
    }

    [HttpGet]
    public IActionResult AddAdmin()
    {
        return View();
    }

    [HttpPost]
    public IActionResult AddAdmin(Admin p)
    {
        am.AddAdmin(p);
        return RedirectToAction("AdminIndex", "Admin");
    }
}

```

```

[HttpGet]
public IActionResult AddTeam()
{
    List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                        select new SelectListItem
                                        {
                                            Text = x.CountryName,
                                            Value = x.CountryID.ToString()
                                        }
                                        ).ToList();

    ViewBag.country = valuecountry;

    List<SelectListItem> valueleague = (from x in lm.GetList()
                                        select new SelectListItem
                                        {
                                            Text = x.LeagueName,
                                            Value = x.LeagueID.ToString()
                                        }
                                        ).ToList();

    ViewBag.league = valueleague;

    return View();
}

[HttpPost]
public IActionResult AddTeam(Team p)
{
    tm.AddTeam(p);
    return RedirectToAction("TeamIndex", "Admin");
}

[HttpGet]
public IActionResult AddLeague()
{
    return View();
}

[HttpPost]
public IActionResult AddLeague(League p)
{
    lm.AddLeague(p);
    return RedirectToAction("LeagueIndex", "Admin");
}

[HttpGet]
public IActionResult AddBlog()
{
    List<SelectListItem> valueblog = (from x in wm.GetList()
                                        select new SelectListItem
                                        {
                                            Text = x.WriterName + " " +
x.WriterSurname,
                                            Value = x.WriterID.ToString()
                                        }
                                        ).ToList();

    ViewBag.blog = valueblog;

    return View();
}

[HttpPost]
public IActionResult AddBlog(Blog p, AddBlogImage i)
{

```

```

        if (i.BlogImage != null)
        {
            var extension = Path.GetExtension(i.BlogImage.FileName);
            var newimagename = Guid.NewGuid() + extension;
            var location = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot/BlogImageFiles/", newimagename);
            var stream = new FileStream(location, FileMode.Create);
            i.BlogImage.CopyTo(stream);
            p.BlogImage = newimagename;
        }

        p.BlogStatus = true;
        blgm.AddBlog(p);
        return RedirectToAction("BlogIndex", "Admin");
    }

    [HttpGet]
    public IActionResult AddWriter()
    {
        List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                             select new SelectListItem
                                             {
                                                 Text = x.CountryName,
                                                 Value = x.CountryID.ToString()
                                             }).ToList();

        ViewBag.country = valuecountry;

        return View();
    }

    [HttpPost]
    public IActionResult AddWriter(Writer p, AddProfileImage i)
    {
        if (i.WriterImage != null)
        {
            var extension = Path.GetExtension(i.WriterImage.FileName);
            var newimagename = Guid.NewGuid() + extension;
            var location = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot/WriterImageFiles/", newimagename);
            var stream = new FileStream(location, FileMode.Create);
            i.WriterImage.CopyTo(stream);
            p.WriterImage = newimagename;
        }

        p.WriterCreateDate = DateTime.Parse(DateTime.Now.ToShortDateString());
        p.WriterStatus = true;
        wm.AddWriter(p);
        return RedirectToAction("WriterIndex", "Admin");
    }

    [HttpGet]
    public IActionResult UpdatePlayer(int id)
    {
        List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                             select new SelectListItem
                                             {
                                                 Text = x.CountryName,
                                                 Value = x.CountryID.ToString()
                                             }).ToList();

        ViewBag.country = valuecountry;
    }

```

```

List<SelectListItem> valueteam = (from x in tm.GetList()
                                select new SelectListItem
                                {
                                    Text = x.TeamName,
                                    Value = x.TeamID.ToString()
                                }
                                ).ToList();

ViewBag.team = valueteam;

List<SelectListItem> valueleague = (from x in lm.GetList()
                                    select new SelectListItem
                                    {
                                        Text = x.LeagueName,
                                        Value = x.LeagueID.ToString()
                                    }
                                    ).ToList();

ViewBag.league = valueleague;

List<SelectListItem> valueskill = (from x in sm.GetList()
                                   select new SelectListItem
                                   {
                                       Text = x.SkillPlayerName,
                                       Value = x.SkillID.ToString()
                                   }
                                   ).ToList();

ViewBag.skill = valueskill;

var values = pm.GeyByID(id);
return View(values);
}

[HttpPost]
public IActionResult UpdatePlayer(Player p)
{
    p.PlayerStatus = true;
    pm.UpdatePlayer(p);
    return RedirectToAction("Index", "Admin");
}

[HttpGet]
public IActionResult UpdateSkill(int id)
{
    var values = pm.GeyByID(id);
    return View(values);
}

[HttpPost]
public IActionResult UpdateAdmin(Admin p)
{
    var list = new List<SelectListItem>
    {
        new SelectListItem{ Text="A", Value = "A" },
        new SelectListItem{ Text="B", Value = "B" },
        new SelectListItem{ Text="C", Value = "C", Selected = true },
    };

    ViewBag.role = list;

    p.AdminStatus = true;
    am.UpdateAdmin(p);
    return RedirectToAction("AdminIndex", "Admin");
}

```

```

    }

    [HttpGet]
    public IActionResult UpdateAdmin(int id)
    {
        var values = am.GetByID(id);
        return View(values);
    }

    [HttpPost]
    public IActionResult UpdateSkill(Skill p)
    {
        sm.UpdateSkill(p);
        return RedirectToAction("SkillIndex", "Admin");
    }

    [HttpGet]
    public IActionResult UpdateTeam(int id)
    {
        List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                             select new SelectListItem
                                             {
                                                 Text = x.CountryName,
                                                 Value = x.CountryID.ToString()
                                             }).ToList();

        ViewBag.country = valuecountry;

        List<SelectListItem> valueleague = (from x in lm.GetList()
                                             select new SelectListItem
                                             {
                                                 Text = x.LeagueName,
                                                 Value = x.LeagueID.ToString()
                                             }).ToList();

        ViewBag.league = valueleague;

        var values = tm.GetByID(id);
        return View(values);
    }

    [HttpPost]
    public IActionResult UpdateTeam(Team p)
    {
        tm.UpdateTeam(p);
        return RedirectToAction("TeamIndex", "Admin");
    }

    [HttpGet]
    public IActionResult UpdateLeague(int id)
    {
        var values = lm.GetByID(id);
        return View(values);
    }

    [HttpPost]
    public IActionResult UpdateLeague(League p)
    {
        lm.UpdateLeague(p);
        return RedirectToAction("LeagueIndex", "Admin");
    }

```

```

[HttpGet]
public IActionResult UpdateBlog(int id)
{
    var c = new Context();
    List<SelectListItem> valueblog = (from x in wm.GetList()
                                     select new SelectListItem
                                     {
                                         Text = x.WriterName + " " +
                                         Value = x.WriterID.ToString()
                                     }
                                     ).ToList();

    ViewBag.blog = valueblog;
    ViewBag.pp = c.Blogs.Where(x => x.BlogID == id).Select(y =>
y.BlogImage).FirstOrDefault();
    var values = blgm.GetByID(id);
    return View(values);
}
[HttpPost]
public IActionResult UpdateBlog(Blog p)
{
    p.BlogStatus = true;
    blgm.UpdateBlog(p);
    return RedirectToAction("BlogIndex", "Admin");
}

[HttpGet]
public IActionResult UpdateWriter(int id)
{
    var c = new Context();
    List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                         select new SelectListItem
                                         {
                                             Text = x.CountryName,
                                             Value = x.CountryID.ToString()
                                         }
                                         ).ToList();

    ViewBag.country = valuecountry;

    ViewBag.pp = c.Writers.Where(x => x.WriterID == id).Select(y =>
y.WriterImage).FirstOrDefault();

    var values = wm.GetByID(id);
    return View(values);
}

[HttpPost]
public IActionResult UpdateWriter(Writer p)
{
    p.WriterCreateDate = DateTime.Parse(DateTime.Now.ToShortDateString());
    p.WriterStatus = true;
    wm.UpdateWriter(p);
    return RedirectToAction("WriterIndex", "Admin");
}

public IActionResult DeletePlayer(int id)
{
    var playervalue = pm.GeyByID(id);
    pm.DeletePlayer(playervalue);
    return RedirectToAction("Index", "Admin");
}

```

```

public IActionResult DeleteSkill(int id)
{
    var skillvalue = sm.GetByID(id);
    sm.DeleteSkill(skillvalue);
    return RedirectToAction("SkillIndex", "Admin");
}

public IActionResult DeleteTeam(int id)
{
    var teamvalue = tm.GetByID(id);
    tm.DeleteTeam(teamvalue);
    return RedirectToAction("TeamIndex", "Admin");
}

public IActionResult DeleteLeague(int id)
{
    var leaguevalue = lm.GetByID(id);
    lm.DeleteLeague(leaguevalue);
    return RedirectToAction("LeagueIndex", "Admin");
}

public IActionResult DeleteBlog(int id)
{
    var blogvalue = blgm.GetByID(id);
    blgm.DeleteBlog(blogvalue);
    return RedirectToAction("BlogIndex", "Admin");
}

public IActionResult DeleteWriter(int id)
{
    var writervalue = wm.GetByID(id);
    wm.DeleteWriter(writervalue);
    return RedirectToAction("WriterIndex", "Admin");
}

public IActionResult DeleteAdmin(int id)
{
    var adminvalue = am.GetByID(id);
    am.DeleteAdmin(adminvalue);
    return RedirectToAction("AdminIndex", "Admin");
}
}

public class BlogController : Controller
{
    BlogManager bm = new BlogManager(new EfBlogRepository());

    [HttpGet]
    public IActionResult Index()
    {
        var values = bm.GetListWithAllAttribution();
        return View(values);
    }

    [HttpPost]
    public IActionResult Index(Blog p, AddBlogImage i)
    {
        if (i.BlogImage != null)
        {
            var extension = Path.GetExtension(i.BlogImage.FileName);
            var newimagename = Guid.NewGuid() + extension;

```



```

        var location = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot/BlogImageFiles/", newimagename);
        var stream = new FileStream(location, FileMode.Create);
        i.BlogImage.CopyTo(stream);
        p.BlogImage = newimagename;
    }

    p.BlogStatus = true;
    p.BlogThumbnailImage = "";
    p.BlogCreateDate = DateTime.Parse(DateTime.Now.ToShortDateString());
    p.WriterID = 5;
    bm.AddBlog(p);
    return RedirectToAction("Index", "Blog");
}

public IActionResult BlogReadAll(int id)
{
    ViewBag.i = id;
    var values = bm.GetListWithAllAttribution2(id);
    return View(values);
}

}

public class CommentController : Controller
{
    CommentManager cm = new CommentManager(new EfCommentRepository());
    public IActionResult Index()
    {
        return View();
    }

    public PartialViewResult CommentListByBlog(int id)
    {
        var values = cm.GetList(id);
        return PartialView(values);
    }
}

public class Contact : Controller
{
    ContactManager cm = new ContactManager(new EfContactRepository());

    [HttpGet]
    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Index(Contactt p)
    {
        p.ContactDate = DateTime.Parse(DateTime.Now.ToShortDateString());
        p.ContactStatus = true;
        cm.AddContact(p);
        return RedirectToAction("Index", "Blog");
    }
}

```

```

public IActionResult Error1(int code)
{
    return View();
}
public class LeagueController : Controller
{
    LeagueManager lm = new LeagueManager(new EfLeagueRepository());
    public IActionResult Index()
    {
        var values = lm.GetList();
        return View(values);
    }
}
public class Login : Controller
{
    [AllowAnonymous]
    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    [AllowAnonymous]
    public async Task<IActionResult> Index(Writer p)
    {
        LoginValidator lv = new LoginValidator();
        ValidationResult results = lv.Validate(p);

        var c = new Context();
        var datavalue = c.Writers.FirstOrDefault(x => x.WriterUserName ==
p.WriterUserName && x.WriterPassword == p.WriterPassword);

        if (results.IsValid)
        {
            if (datavalue != null)
            {
                //HttpContext.Session.SetString("WriterUserName",p.WriterUserName);

                var claims = new List<Claim>
                {
                    new Claim(ClaimTypes.Name,p.WriterUserName)
                };

                var useridentity = new ClaimsIdentity(claims, "a");
                ClaimsPrincipal principal = new ClaimsPrincipal(useridentity);
                await HttpContext.SignInAsync(principal);
                return RedirectToAction("Index", "WriterProfile");
            }
        }
        else
        {
            foreach (var item in results.Errors)
            {
                ModelState.AddModelError(item.PropertyName, item.ErrorMessage);
            }
        }

        return View();
    }
}

```

```

        public async Task<IActionResult> Logout()
        {
            await
HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
            return RedirectToAction("Index", "Login");
        }
    }
}

public class PlayerController : Controller
{
    PlayerManager pm = new PlayerManager(new EfPlayerRepository());
    public IActionResult Index()
    {
        var values = pm.GetListWithAllAttribution();
        return View(values);
    }

    public IActionResult GetMostValuablePlayer()
    {
        var c = new Context();
        var values = pm.GetListWithAllAttribution().OrderByDescending(x =>
x.PlayerValue).Take(3);
        return View(values);
    }

    public IActionResult PlayerDetails(int id)
    {
        var c = new Context();
        ViewBag.i = id;
        ViewBag.pn = c.Players.Find(id).PlayerName;
        ViewBag.ps = c.Players.Find(id).PlayerSurname;
        var values = pm.GetListWithAllAttribution2(id);
        return View(values);
    }
}

public class RegisterController : Controller
{
    WriterManager wm = new WriterManager(new EfWriterRepository());
    CountryManager cm = new CountryManager(new EfCountryRepository());

    [AllowAnonymous]
    [HttpGet]
    public IActionResult Index()
    {
        List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
select new SelectListItem
{
    Text = x.CountryName,
    Value = x.CountryID.ToString()
}).ToList();

        ViewBag.country = valuecountry;

        return View();
    }

    [AllowAnonymous]
    [HttpPost]

```

```

public IActionResult Index(Writer p, AddProfileImage i)
{
    RegisterValidator rv = new RegisterValidator();
    ValidationResult results = rv.Validate(p);
    if (results.IsValid)
    {
        if (i.WriterImage != null)
        {
            var extension = Path.GetExtension(i.WriterImage.FileName);
            var newimagename = Guid.NewGuid() + extension;
            var location = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot/WriterImageFiles/", newimagename);
            var stream = new FileStream(location, FileMode.Create);
            i.WriterImage.CopyTo(stream);
            p.WriterImage = newimagename;
        }

        p.WriterStatus = true;
        p.WriterAbout = "";
        p.WriterCreateDate = DateTime.Parse(DateTime.Now.ToShortDateString());
        p.WriterAddress = "";
        wm.AddWriter(p);
        return RedirectToAction("Index", "Login");
    }
    else
    {
        foreach (var item in results.Errors)
        {
            ModelState.AddModelError(item.PropertyName, item.ErrorMessage);
        }
    }

    List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
                                        select new SelectListItem
                                        {
                                            Text = x.CountryName,
                                            Value = x.CountryID.ToString()
                                        }).ToList();

    ViewBag.country = valuecountry;

    return View();
}
}
public class SSSController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
}
public class TeamController : Controller
{
    TeamManager tm = new TeamManager(new EfTeamRepository());
    public IActionResult Index()
    {
        var values = tm.GetListWithAllAttribution();
        return View(values);
    }
}
}

```

```

public class WriterProfile : Controller
{
    BlogManager bm = new BlogManager(new EfBlogRepository());
    WriterManager wm = new WriterManager(new EfWriterRepository());
    CountryManager cm = new CountryManager(new EfCountryRepository());

    [Authorize]
    [HttpGet]
    public IActionResult Index(int id)
    {
        var username = User.Identity.Name;
        ViewBag.un = username;

        var c = new Context();

        var name = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterName).FirstOrDefault();
        var surname = c.Writers.Where(x => x.WriterUserName == username).Select(y
=> y.WriterSurname).FirstOrDefault();
        var tt = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterTitle).FirstOrDefault();
        ViewBag.i = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterID).FirstOrDefault();
        var pp = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterImage).FirstOrDefault();

        ViewBag.pp = pp;
        ViewBag.tt = tt;
        ViewBag.n = name;
        ViewBag.s = surname;

        var values = bm.GetListWithAllAttribution2(id);
        return View(values);
    }

    [HttpPost]
    public IActionResult Index(Blog p, AddBlogImage i, int id)
    {
        if (i.BlogImage != null)
        {
            var extension = Path.GetExtension(i.BlogImage.FileName);
            var newimagename = Guid.NewGuid() + extension;
            var location = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot/BlogImageFiles/", newimagename);
            var stream = new FileStream(location, FileMode.Create);
            i.BlogImage.CopyTo(stream);
            p.BlogImage = newimagename;
        }

        var username = User.Identity.Name;
        ViewBag.un = username;

        var c = new Context();

        var name = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterName).FirstOrDefault();
        var surname = c.Writers.Where(x => x.WriterUserName == username).Select(y
=> y.WriterSurname).FirstOrDefault();
        var tt = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterTitle).FirstOrDefault();

```

```

        ViewBag.i = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterID).FirstOrDefault();
        var pp = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterImage).FirstOrDefault();

        ViewBag.pp = pp;
        ViewBag.tt = tt;
        ViewBag.n = name;
        ViewBag.s = surname;

        p.BlogStatus = true;
        p.BlogThumbnailImage = "";
        p.BlogCreateDate = DateTime.Parse(DateTime.Now.ToShortDateString());
        p.WriterID = ViewBag.i;

        bm.AddBlog(p);

        var values = bm.GetListWithAllAttribution2(id);
        return View(values);
    }

    public IActionResult WriterProfiles(int id)
    {
        var c = new Context();
        var username = c.Writers.Find(id).WriterName;
        var surname = c.Writers.Find(id).WriterSurname;
        var title = c.Writers.Find(id).WriterTitle;
        var pp = c.Writers.Find(id).WriterImage;

        ViewBag.pp = pp;
        ViewBag.un = username;
        ViewBag.sn = surname;
        ViewBag.tt = title;
        ViewBag.i = id;
        var values = bm.GetListWithAllAttribution2(id);
        return View(values);
    }

    [Authorize]
    [HttpGet]
    public IActionResult WriterSettings(int id)
    {
        var username = User.Identity.Name;
        ViewBag.un = username;

        var c = new Context();

        var name = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterName).FirstOrDefault();
        var surname = c.Writers.Where(x => x.WriterUserName == username).Select(y
=> y.WriterSurname).FirstOrDefault();
        var tt = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterTitle).FirstOrDefault();
        ViewBag.i = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterID).FirstOrDefault();
        var pp = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterImage).FirstOrDefault();
        var un = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterUserName).FirstOrDefault();

```

```

        var ct = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.Country.CountryName).FirstOrDefault();
        var ab = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterAbout).FirstOrDefault();
        var ad = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterAddress).FirstOrDefault();
        var ml = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterMail).FirstOrDefault();
        var dt = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterBirthDate).FirstOrDefault();
        var pass = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterPassword).FirstOrDefault();

        var s = (((DateTime)dt).ToString("dd.MM.yyyy"));

        ViewBag.pass = pass;
        ViewBag.dt = s;
        ViewBag.ab = ab;
        ViewBag.ad = ad;
        ViewBag.ml = ml;
        ViewBag.ct = ct;
        ViewBag.un = un;
        ViewBag.pp = pp;
        ViewBag.tt = tt;
        ViewBag.n = name;
        ViewBag.s = surname;

        List<SelectListItem> valuecountry = (from x in cm.GetListCountry()
select new SelectListItem
{
    Text = x.CountryName,
    Value = x.CountryID.ToString()
}).ToList();

        ViewBag.country = valuecountry;

        var values = wm.GetByID(ViewBag.i);
        return View(values);
    }

    [HttpPost]
    public IActionResult WriterSettings(Writer p)
    {
        var username = User.Identity.Name;
        ViewBag.un = username;

        var c = new Context();
        var pp = c.Writers.Where(x => x.WriterUserName == username).Select(y =>
y.WriterImage).FirstOrDefault();

        ViewBag.pp = pp;
        p.WriterImage = pp;

        wm.UpdateWriter(p);
        return RedirectToAction("WriterSettings", "WriterProfile");
    }
}

```

Models

```
public class AddBlogImage
{
    public int BlogID { get; set; }
    public string BlogTitle { get; set; }
    public string BlogContent { get; set; }
    public string BlogThumbnailImage { get; set; }
    public IFormFile BlogImage { get; set; }
    public DateTime BlogCreateDate { get; set; }
    public bool BlogStatus { get; set; }
    public int WriterID { get; set; }
}
public class AddProfileImage
{
    public int WriterID { get; set; }
    public string WriterUserName { get; set; }
    public string WriterName { get; set; }
    public string WriterSurname { get; set; }
    public IFormFile WriterImage { get; set; }
    public string WriterMail { get; set; }
    public string WriterPassword { get; set; }
    public string WriterAbout { get; set; }
    public string WriterTitle { get; set; }
    public DateTime WriterBirthDate { get; set; }
    public DateTime WriterCreateDate { get; set; }
    public string WriterAddress { get; set; }
    public int CountryID { get; set; }
    public bool WriterStatus { get; set; }
}
```


ViewComponents

```
public class WriterLastBlog : ViewComponent
{
    BlogManager bm = new BlogManager(new EfBlogRepository());
    public IViewComponentResult Invoke(int id)
    {
        var values = bm.GetBlogListByWriter(id);
        return View(values);
    }
}

public class CommentListByBlog : ViewComponent
{
    CommentManager cm = new CommentManager(new EfCommentRepository());
    public IViewComponentResult Invoke(int id)
    {
        var values = cm.GetList(id);
        return View(values);
    }
}

public class GetMostValuablePlayer : ViewComponent
{
    PlayerManager pm = new PlayerManager(new EfPlayerRepository());
    public IViewComponentResult Invoke()
    {
        var values = pm.GetListWithAllAttribution().OrderByDescending(x =>
x.PlayerValue).Take(3).ToList();
        return View(values);
    }
}

public class WriterProfileLastBlog : ViewComponent
{
    BlogManager bm = new BlogManager(new EfBlogRepository());
    public IViewComponentResult Invoke(int id)
    {
        var values = bm.GetBlogListByWriter(id);
        return View(values);
    }
}
```

Views : Admin

```
@model EntityLayer.Concrete.Admin
@{
    ViewData["Title"] = "AddAdmin";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Admin Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddAdmin", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Kullanıcı
                        Adı</label>

                        <div class="col-sm-3"><input type="text" name="AdminUserName"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Şifresi</label>

                        <div class="col-sm-3"><input type="password"
name="AdminPassword" class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Rolü</label>

                        <div class="col-sm-3">
                            <select name="AdminRole" class="form-control m-b">
                                <option value="A">A</option>
                                <option value="B">B</option>
                                <option value="C">C</option>
                            </select>
                        </div>
                    </div>
                    <div class="form-group row">
                        <div class="col-sm-4 col-sm-offset-2">
                            <button class="btn btn-primary btn-sm" type="submit">Admin
Ekle</button>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>
```

```

@model EntityLayer.Concrete.Blog
@{
    ViewData["Title"] = "AddBlog";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}
<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Blog Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddBlog", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Blog Başlığı</label>

                        <div class="col-sm-3"><input type="text" name="BlogTitle"
class="form-control"></div>
                        <div class="hr-line-dashed"></div>
                        <div class="form-group row">
                            <label class="col-sm-2 col-form-label">Blog Alanı</label>

                            <div class="col-sm-3"><input type="text" name="BlogContent"
class="form-control"></div>
                            <div class="hr-line-dashed"></div>
                            <div class="form-group row">
                                <label class="col-sm-2 col-form-label">Resim</label>

                                <div class="col-sm-3"><input type="file" name="BlogImage"
class="form-control"></div>
                                <div class="hr-line-dashed"></div>
                                <div class="form-group row">
                                    <label class="col-sm-2 col-form-label">Blog Tarihi</label>

                                    <div class="col-sm-3"><input type="text" name="BlogCreateDate"
class="form-control"></div>
                                    <div class="hr-line-dashed"></div>
                                    <div class="form-group row">
                                        <label class="col-sm-2 col-form-label">Yazar</label>

                                        <div class="col-sm-3">@Html.DropDownListFor(x => x.WriterID,
(List<SelectListItem>)ViewBag.blog, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
                                        <div class="hr-line-dashed"></div>
                                        <div class="form-group row">
                                            <div class="col-sm-4 col-sm-offset-2">
                                                <button class="btn btn-primary btn-sm" type="submit">Blog
Ekle</button>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>

```

```

</div>
@model EntityLayer.Concrete.League
@{
    ViewData["Title"] = "AddLeague";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>lig Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddLeague", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Lig Adı</label>

                        <div class="col-sm-3"><input type="text" name="LeagueName"
class="form-control"></div>
                        <div class="hr-line-dashed"></div>
                        <div class="form-group row">
                            <label class="col-sm-2 col-form-label">Değeri</label>

                            <div class="col-sm-3"><input type="text" name="LeagueValue"
class="form-control"></div>
                            <div class="hr-line-dashed"></div>
                            <div class="form-group row">
                                <label class="col-sm-2 col-form-label">Takım Sayısı</label>

                                <div class="col-sm-3"><input type="text"
name="LeagueTeamCount" class="form-control"></div>
                                <div class="hr-line-dashed"></div>
                                <div class="form-group row">
                                    <label class="col-sm-2 col-form-label">Kuruluş Yılı</label>

                                    <div class="col-sm-3"><input type="text"
name="LeagueFoundationYear" class="form-control"></div>
                                    <div class="hr-line-dashed"></div>
                                    <div class="form-group row">
                                        <label class="col-sm-2 col-form-label">Açıklama</label>

                                        <div class="col-sm-3"><input type="text"
name="LeagueDescription" class="form-control"></div>
                                        <div class="hr-line-dashed"></div>
                                        <div class="form-group row">
                                            <div class="col-sm-4 col-sm-offset-2">
                                                <button class="btn btn-primary btn-sm" type="submit">Lig
Ekle</button>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>

```

```
</div>
```

```
@model EntityLayer.Concrete.Player
```

```
@{
```

```
    ViewData["Title"] = "AddPlayer";
```

```
    Layout = "~/Views/Shared/AdminLayout.cshtml";
```

```
}
```

```
<div class="row">
```

```
    <div class="col-lg-12">
```

```
        <div class="ibox ">
```

```
            <div class="ibox-title">
```

```
                <h5>Futbolcu Ekleme </h5>
```

```
            </div>
```

```
            <div class="ibox-content">
```

```
                @using (Html.BeginForm("AddPlayer", "Admin", FormMethod.Post, new {  
@class = "", enctype = "multipart/form-data" }))
```

```
                {
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Adı</label>
```

```
                        <div class="col-sm-3"><input type="text" name="PlayerName"  
class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Soyadı</label>
```

```
                        <div class="col-sm-3"><input type="text" name="PlayerSurname"  
class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Yaş</label>
```

```
                        <div class="col-sm-3"><input type="text" name="PlayerAge"  
class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Boy</label>
```

```
                        <div class="col-sm-3"><input type="text" name="PlayerHeight"  
class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Kilo</label>
```

```
                        <div class="col-sm-3"><input type="text" name="PlayerWeight"  
class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Pozisyon ISO</label>
```

```
                        <div class="col-sm-3"><input type="text"  
name="PlayerPositionISO" class="form-control"></div>
```

```
                    </div>
```

```
                    <div class="hr-line-dashed"></div>
```

```
                    <div class="form-group row">
```

```
                        <label class="col-sm-2 col-form-label">Pozisyon</label>
```

```

        <div class="col-sm-3"><input type="text" name="PlayerPosition"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Açıklama</label>

        <div class="col-sm-3"><input type="text"
name="PlayerDescription" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Kulüp</label>

        <div class="col-sm-3"><input type="text" name="PlayerClub"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Doğum Tarihi</label>

        <div class="col-sm-3"><input type="datetime"
name="PlayerBirthDate" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Menajer</label>

        <div class="col-sm-3"><input type="text" name="PlayerManager"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Değer</label>

        <div class="col-sm-3"><input type="text" name="PlayerValue"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Numara</label>

        <div class="col-sm-3"><input type="text" name="PlayerNumber"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Ülke</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Takım</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.TeamID,
(List<SelectListItem>)ViewBag.team, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>

```

```

<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Lig</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.LeagueID,
(List<SelectListItem>)ViewBag.league, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Yetenek</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.SkillID,
(List<SelectListItem>)ViewBag.skill, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
</div>
<div class="hr-line-dashed"></div>

<div class="form-group row">
    <div class="col-sm-4 col-sm-offset-2">
        <button class="btn btn-primary btn-sm"
type="submit">Oyuncuyu Ekle</button>
    </div>
</div>
}
</div>
</div>
</div>
</div>

```

@model EntityLayer.Concrete.Skill

```

@{
    ViewData["Title"] = "AddSkill";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

```

```

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Yetenek Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddSkill", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Gol</label>

                        <div class="col-sm-3"><input type="text" name="Goal"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Asist</label>

                        <div class="col-sm-3"><input type="text" name="Assist"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">BBC</label>

```

```

        <div class="col-sm-3"><input type="text" name="BBC"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">AnahtarPas</label>

        <div class="col-sm-3"><input type="text" name="KeyPass"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">İsabetli Şut</label>

        <div class="col-sm-3"><input type="text" name="ShootsOnTarget"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">İsabetsiz Şut</label>

        <div class="col-sm-3"><input type="text"
name="ShootsOffTarget" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Kırmızı Kart</label>

        <div class="col-sm-3"><input type="text" name="RedCards"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Sarı Kart</label>

        <div class="col-sm-3"><input type="text" name="YellowCards"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Başarısız
Dripling</label>

        <div class="col-sm-3"><input type="text"
name="UnsuccesfulDribbling" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Başarılı
Dripling</label>

        <div class="col-sm-3"><input type="datetime"
name="SuccesfulDribbling" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Kazanılan İkili
Mücadele</label>

        <div class="col-sm-3"><input type="text" name="AerialDuelsWon"
class="form-control"></div>

```



```

        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Kaybedilen ikili
Mücadele</label>

            <div class="col-sm-3"><input type="text"
name="AerialDuelsLost" class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Pozisyon Kaybı</label>

            <div class="col-sm-3"><input type="text" name="PossesionLost"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Kazanılan
Pozisyon</label>

            <div class="col-sm-3"><input type="text" name="PossesionWon"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Penaltı</label>

            <div class="col-sm-3"><input type="text" name="Penalties"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Uzun Top</label>

            <div class="col-sm-3"><input type="text" name="LongBall"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Yedek Maç
Sayısı</label>

            <div class="col-sm-3"><input type="text" name="BenchMatch"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">İlk 11</label>

            <div class="col-sm-3"><input type="text" name="StartedMatch"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Reyting</label>

            <div class="col-sm-3"><input type="text" name="Rating"
class="form-control"></div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">

```

```

        <label class="col-sm-2 col-form-label">MOTM</label>

        <div class="col-sm-3"><input type="text" name="MOTM"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Yetenek Futbolcu
Adı</label>

        <div class="col-sm-3"><input type="text"
name="SkillPlayerName" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <div class="col-sm-4 col-sm-offset-2">
            <button class="btn btn-primary btn-sm"
type="submit">Yetenek Ekle</button>
        </div>
    </div>
}
</div>
</div>
</div>
</div>
}

@model EntityLayer.Concrete.Team
@{
    ViewData["Title"] = "AddTeam";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Takım Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddTeam", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Takım Adı</label>

                        <div class="col-sm-3"><input type="text" name="TeamName"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Teknik Direktör</label>

                        <div class="col-sm-3"><input type="text" name="TeamCoach"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Takım Değeri</label>

                        <div class="col-sm-3"><input type="text" name="TeamValue"
class="form-control"></div>
                    </div>
                }
            </div>
        </div>
    </div>

```

```

<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Stat Adı</label>

    <div class="col-sm-3"><input type="text" name="TeamStatName"
class="form-control"></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Futbolcu Sayısı</label>

    <div class="col-sm-3"><input type="text"
name="TeamPlayerCount" class="form-control"></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Açıklama</label>

    <div class="col-sm-3"><input type="text"
name="TeamDescription" class="form-control"></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Kuruluş Yılı</label>

    <div class="col-sm-3"><input type="text"
name="TeamFoundationYear" class="form-control"></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Logo</label>

    <div class="col-sm-3"><input type="text" name="TeamImage"
class="form-control"></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Ülke</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Lig</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.LeagueID,
(List<SelectListItem>)ViewBag.league, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
</div>
<div class="hr-line-dashed"></div>

<div class="form-group row">
    <div class="col-sm-4 col-sm-offset-2">
        <button class="btn btn-primary btn-sm" type="submit">Takım
Ekle</button>
    </div>
</div>
}
</div>
</div>

```

```

    </div>
</div>
@model EntityLayer.Concrete.Writer
@{
    ViewData["Title"] = "AddWriter";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Yazar Ekleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("AddWriter", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Kullanıcı Adı</label>

                        <div class="col-sm-3"><input type="text" name="WriterUserName"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Adı</label>

                        <div class="col-sm-3"><input type="text" name="WriterName"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Soyadı</label>

                        <div class="col-sm-3"><input type="text" name="WriterSurname"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Resim</label>

                        <div class="col-sm-3"><input type="file" name="WriterImage"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Mail</label>

                        <div class="col-sm-3"><input type="text" name="WriterMail"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Şifre</label>

                        <div class="col-sm-3"><input type="text" name="WriterPassword"
class="form-control"></div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Hakkında</label>

```

```

        <div class="col-sm-3"><input type="text" name="WriterAbout"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Ünvan</label>

        <div class="col-sm-3"><input type="text" name="WriterTitle"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Doğum Tarihi</label>

        <div class="col-sm-3"><input type="datetime"
name="WriterBirthDate" class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Adres</label>

        <div class="col-sm-3"><input type="text" name="WriterAddress"
class="form-control"></div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Ülke</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <div class="col-sm-4 col-sm-offset-2">
            <button class="btn btn-primary btn-sm" type="submit">Yazar
Ekle</button>
        </div>
    </div>
    </div>
}
</div>
</div>
</div>
</div>
@using EntityLayer.Concrete
@model List<Admin>
@{
    ViewData["Title"] = "AdminIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Admins CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>

```

```

        <tr>
            <th>#</th>
            <th>Admin Kullanıcı Adı</th>
            <th>Admin Şifre</th>
            <th>Admin Rolü</th>
            <th>Admin Durumu</th>
            <th>Güncelle</th>
            <th>Sil</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@item.AdminID</td>
                <td>@item.AdminUserName</td>
                <td>@item.AdminPassword</td>
                <td>@item.AdminRole</td>
                <td>@item.AdminStatus</td>
                <td><a class="btn btn-primary"
href="/Admin/UpdateAdmin/@item.AdminID/"><span class="fa fa-check"></span></a></td>
                <td><a class="btn btn-danger"
onclick="confirmation(event, '@item.AdminUserName')"
href="/Admin/DeleteAdmin/@item.AdminID/"><span class="fa fa-trash"></span></a></td>
            </tr>
        }
    </tbody>
</table>
<a class="btn btn-info" style="margin-top:10px;" asp-action="AddAdmin"
asp-controller="Admin">Admin Ekle</a>
</div>
</div>
</div>
@model EntityLayer.Concrete.Admin
@{
    Layout = null;
}

<!DOCTYPE html>
<html>

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin | Login</title>

    <link href="~/inspinia-gh-pages/css/bootstrap.min.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/font-awesome/css/font-awesome.css"
rel="stylesheet">

    <link href="~/inspinia-gh-pages/css/animate.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/style.css" rel="stylesheet">

</head>

<body class="gray-bg">

    <div class="middle-box text-center loginscreen animated fadeInDown">
        <div>

```

```

        <div>

            <h1 class="logo-name">SSN</h1>

        </div>
        <h3>Admin Login</h3>
        @using (Html.BeginForm("AdminLoginIndex", "Admin", FormMethod.Post, new {
@class = "m-t" })))
        {
            <div class="form-group">
                <input type="text" name="AdminUserName" class="form-control"
placeholder="Kullanıcı Adı">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.AdminUserName)</p>
            </div>
            <div class="form-group">
                <input type="password" name="AdminPassword" class="form-control"
placeholder="Şifre">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.AdminPassword)</p>
            </div>
            <button type="submit" class="btn btn-primary block full-width m-
b">Login</button>
        }
    </div>
</div>

<!-- Mainly scripts -->
<script src="~/inspinia-gh-pages/js/jquery-3.1.1.min.js"></script>
<script src="~/inspinia-gh-pages/js/popper.min.js"></script>
<script src="~/inspinia-gh-pages/js/bootstrap.js"></script>

</body>

</html>
@using EntityLayer.Concrete
@model List<Blog>

@{
    ViewData["Title"] = "BlogIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Blogs CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Blog Başlığı</th>
                            <th>Blog Alanı</th>
                            <th>Küçük Resim</th>
                            <th>Büyük Resim</th>
                            <th>Blog Tarihi</th>
                            <th>Blog Durumu</th>
                            <th>Blog Yazarı</th>
                        </tr>
                    </thead>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

                <th>Güncelle</th>
                <th>Sil</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>@item.BlogID</td>
                    <td>@item.BlogTitle</td>
                    <td>@item.BlogContent</td>
                    <td></td>
                    <td><a href="/BlogImageFiles/@item.BlogImage">Resimi
Gör</a></td>

                    <td>@(((DateTime)item.BlogCreateDate).ToString("dd.MM.yyyy"))</td>
                    <td>@item.BlogStatus</td>
                    <td>@item.Writer.WriterName
@item.Writer.WriterSurname</td>
                    <td><a class="btn btn-primary"
href="/Admin/UpdateBlog/@item.BlogID/"><span class="fa fa-check"></span></a></td>
                    <td><a class="btn btn-danger" onclick="confirmation(event,
'@item.BlogTitle')" href="/Admin/DeleteBlog/@item.BlogID/"><span class="fa fa-
trash"></span></a></td>
                </tr>
            }
        </tbody>
    </table>
    <a class="btn btn-info" style="margin-top:10px;" asp-action="AddBlog"
asp-controller="Admin">Blog Ekle</a>
</div>
</div>
</div>
</div>

@using EntityLayer.Concrete
@model List<Comment>
@{
    ViewData["Title"] = "CommentIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Comments CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Yorum Kullanıcı Adı</th>
                            <th>Yorum Başlık</th>
                            <th>Yorum Alanı</th>
                            <th>Yorum Tarihi</th>
                            <th>Durum</th>
                            <th>Gönderi Adı</th>
                        </tr>
                    </thead>
                </table>
            </div>
        </div>
    </div>
</div>

```



```

                <th>Güncelle</th>
                <th>Sil</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>@item.CommentID</td>
                    <td>@item.CommentUserName</td>
                    <td>@item.CommentTitle</td>
                    <td>@item.CommentContent</td>

                    <td>@(((DateTime)item.CommentDate).ToString("dd.MM.yyyy"))</td>
                    <td>@item.CommentStatus</td>
                    <td>@item.Blog.BlogTitle</td>
                    <td><a class="btn btn-primary" href="#"><span
class="fa fa-check"></span></a></td>
                    <td><a class="btn btn-danger"
onclick="confirmation(event, '@item.CommentTitle')" href="#"><span class="fa fa-
trash"></span></a></td>
                </tr>
            }
        </tbody>
    </table>
    <a class="btn btn-info" style="margin-top:10px;" @*asp-
action="AddWriter" asp-controller="Admin"*@>Yorum Ekle</a>
</div>
</div>
</div>
</div>

```

```

@using EntityLayer.Concrete
@model List<Contactt>
@{
    ViewData["Title"] = "ContactIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Contacts CRUD</h5>
            </div>
            <div class="ibox-content">
                <table class="table">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Gönderen</th>
                            <th>Mail Adresi</th>
                            <th>Konu</th>
                            <th>Mesaj</th>
                            <th>Tarih</th>
                            <th>Durum</th>
                            <th>Güncelle</th>
                            <th>Sil</th>
                        </tr>

```

```

        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>@item.ContactID</td>
                    <td>@item.ContactUserName</td>
                    <td>@item.ContactMail</td>
                    <td>@item.ContactSubject</td>
                    <td>@item.ContactMessage</td>

                    <td>@(((DateTime)item.ContactDate).ToString("dd.MM.yyyy"))</td>
                    <td>@item.ContactStatus</td>
                    <td><a class="btn btn-primary"><span class="fa fa-
check"></span></a></td>
                    <td><a class="btn btn-danger"><span class="fa fa-
trash"></span></a></td>
                </tr>
            }
        </tbody>
    </table>

</div>
</div>
</div>
</div>

@using EntityLayer.Concrete
@model List<Country>
@{
    ViewData["Title"] = "CountryIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Countries CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Ülke ISO</th>
                            <th>Ülke Adı</th>
                            <th>Ülke N Adı</th>
                            <th>Ülke ISO3</th>
                            <th>Ülke Num Kodu</th>
                            <th>Ülke Telefon Kodu</th>
                            <th>Güncelle</th>
                            <th>Sil</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model)
                        {
                            <tr>
                                <td>@item.CountryID</td>

```

```
 @item.CountryISO</td>  @item.CountryName</td>  @item.CountryNName</td>  @item.CountryISO3</td>  @item.CountryNumCode</td>  @item.CountryPhoneCode</td>  <a class="btn btn-primary" ><span class="fa fa- check"></span></a></td>  <a class="btn btn-danger" ><span class="fa fa- trash"></span></a></td> </tr> } </tbody> </table> </div> </div> </div> </div> | | | | | | | |
```

```

@using EntityLayer.Concrete
@model List<Player>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Players CRUD</h5>
            </div>
            <div class="ibox-content">
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Adı</th>
                            <th>Soyadı</th>
                            <th>Yaş</th>
                            <th>Boy</th>
                            <th>Kilo</th>
                            <th>Pozisyon ISO</th>
                            <th>Pozisyon</th>
                            <th>Açıklama</th>
                            <th>Kulüp</th>
                            <th>Doğum Tarihi</th>
                            <th>Menajer</th>
                            <th>Değer</th>
                            <th>Numara</th>
                            <th>Durum</th>
                            <th>Ülke</th>
                            <th>Takım</th>
                            <th>Lig</th>
                            <th>Yetenek</th>
                            <th>Güncelle</th>
                            <th>Sil</th>
                        </tr>

```

```

        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>@item.PlayerID</td>
                    <td>@item.PlayerName</td>
                    <td>@item.PlayerSurname</td>
                    <td>@item.PlayerAge</td>
                    <td>@item.PlayerHeight</td>
                    <td>@item.PlayerWeight</td>
                    <td>@item.PlayerPositionISO</td>
                    <td>@item.PlayerPosition</td>
                    <td>@item.PlayerDescription</td>
                    <td>@item.PlayerClub</td>

                    <td>@(((DateTime)item.PlayerBirthDate).ToString("dd.MM.yyyy"))</td>
                    <td>@item.PlayerManager</td>
                    <td>@item.PlayerValue</td>
                    <td>@item.PlayerNumber</td>
                    <td>@item.PlayerStatus</td>
                    <td>@item.Country.CountryName</td>
                    <td>@item.Team.TeamName</td>
                    <td>@item.League.LeagueName</td>
                    <td>@item.Skill.SkillID</td>
                    <td><a class="btn btn-primary"
href="/Admin/UpdatePlayer/@item.PlayerID/"><span class="fa fa-check"></span></a></td>
                    <td><a class="btn btn-danger" id="deletedata"
onclick="confirmation(event, '@item.PlayerName')"
href="/Admin/DeletePlayer/@item.PlayerID/"><span class="fa fa-trash"></span></a></td>
                </tr>
            }
        </tbody>
    </table>
    <a class="btn btn-info" style="margin-top:10px;" asp-
action="AddPlayer" asp-controller="Admin" >Futbolcu Ekle</a>
</div>
</div>
</div>
</div>

@using EntityLayer.Concrete
@model List<League>
@{
    ViewData["Title"] = "LeagueIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Leagues CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Lig Adı</th>

```

```

        <th>Lig Değeri</th>
        <th>Lig Takım Sayısı</th>
        <th>Lig Kuruluş Yılı</th>
        <th>Lig Açıklama</th>
        <th>Güncelle</th>
        <th>Sil</th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>@item.LeagueID</td>
            <td>@item.LeagueName</td>
            <td>@item.LeagueValue</td>
            <td>@item.LeagueTeamCount</td>
            <td>@item.LeagueFoundationYear</td>
            <td>@item.LeagueDescription</td>
            <td><a class="btn btn-primary"
href="/Admin/UpdateLeague/@item.LeagueID/"><span class="fa fa-check"></span></a></td>
            <td><a class="btn btn-danger"
onclick="confirmation(event, '@item.LeagueName')"
href="/Admin/DeleteLeague/@item.LeagueID/"><span class="fa fa-trash"></span></a></td>
        </tr>
    }
</tbody>
</table>
<a class="btn btn-info" style="margin-top:10px;" asp-
action="AddLeague" asp-controller="Admin">Lig Ekle</a>
</div>
</div>
</div>
@using EntityLayer.Concrete
@model List<Skill>
@{
    ViewData["Title"] = "SkillIndex";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Skills CRUD</h5>
            </div>
            <div class="ibox-content">
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Futbolcu Adı</th>
                            <th>Gol</th>
                            <th>Asist</th>
                            <th>BBC</th>
                            <th>Aanahtar Pas</th>
                            <th>İsabetli Şut</th>
                            <th>İsabetsiz Şut</th>
                            <th>Kırmızı Kart</th>
                            <th>Sarı Kart</th>
                        </tr>
                    </thead>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

        <th>Başarısız Dripling</th>
        <th>Başarılı Dripling</th>
        <th>Kazanılan İkili Mücadele</th>
        <th>Kaybedilen İkili Mücadele</th>
        <th>Pozisyon Kaybı</th>
        <th>Pozisyon Kazanma</th>
        <th>Penaltı</th>
        <th>Uzun Top</th>
        <th>Yedek Maç</th>
        <th>İlk 11</th>
        <th>Reyting</th>
        <th>MOTM</th>
        <th>Güncelle</th>
        <th>Sil</th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>@item.SkillID</td>
            <td>@item.SkillPlayerName</td>
            <td>@item.Goal</td>
            <td>@item.Assist</td>
            <td>@item.BBC</td>
            <td>@item.KeyPass</td>
            <td>@item.ShootsOnTarget</td>
            <td>@item.ShootsOffTarget</td>
            <td>@item.RedCards</td>
            <td>@item.YellowCards</td>
            <td>@item.UnsuccessfulDribbling</td>
            <td>@item.SuccesfulDribbling</td>
            <td>@item.AerialDuelsWon</td>
            <td>@item.AerialDuelsLost</td>
            <td>@item.PossesionLost</td>
            <td>@item.PossesionWon</td>
            <td>@item.Penalties</td>
            <td>@item.LongBall</td>
            <td>@item.BenchMatch</td>
            <td>@item.StartedMatch</td>
            <td>@item.Rating</td>
            <td>@item.MOTM</td>
            <td><a class="btn btn-primary"
href="/Admin/UpdateSkill/@item.SkillID/"><span class="fa fa-check"></span></a></td>
            <td><a class="btn btn-danger" onclick="confirmation(event,
'@item.SkillPlayerName')" href="/Admin/DeleteSkill/@item.SkillID/"><span class="fa fa-
trash"></span></a></td>
        </tr>
    }
</tbody>
</table>
<a class="btn btn-info" style="margin-top:10px;" asp-action="AddSkill"
asp-controller="Admin">Yetenek Ekle</a>
</div>
</div>
</div>
</div>
</div>

@using EntityLayer.Concrete
@model List<Team>
@{

```

```

ViewData["Title"] = "TeamIndex";
Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox">
            <div class="ibox-title">
                <h5>Teams CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Takım Adı</th>
                            <th>Teknik Direktör</th>
                            <th>Değer</th>
                            <th>Stat Adı</th>
                            <th>Futbolcu Sayısı</th>
                            <th>Açıklama</th>
                            <th>Kuruluş Yılı</th>
                            <th>Logo</th>
                            <th>Ülke</th>
                            <th>Lig</th>
                            <th>Güncelle</th>
                            <th>Sil</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model)
                        {
                            <tr>
                                <td>@item.TeamID</td>
                                <td>@item.TeamName</td>
                                <td>@item.TeamCoach</td>
                                <td>@item.TeamValue</td>
                                <td>@item.TeamStatName</td>
                                <td>@item.TeamPlayerCount</td>
                                <td>@item.TeamDescription</td>
                                <td>@item.TeamFoundationYear</td>
                                <td>@item.TeamImage</td>
                                <td>@item.Country.CountryName</td>
                                <td>@item.League.LeagueName</td>
                                <td><a class="btn btn-primary"
href="/Admin/UpdateTeam/@item.TeamID/"><span class="fa fa-check"></span></a></td>
                                <td><a class="btn btn-danger" onclick="confirmation(event,
'@item.TeamName')" href="/Admin/DeleteTeam/@item.TeamID/"><span class="fa fa-
trash"></span></a></td>
                            </tr>
                        }

                    </tbody>
                </table>
                <a class="btn btn-info" style="margin-top:10px;" asp-action="Addteam"
asp-controller="Admin">Takım Ekle</a>
            </div>
        </div>
    </div>
</div>

```

```

@model EntityLayer.Concrete.Admin
@{
    ViewData["Title"] = "UpdateAdmin";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Admin Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdateAdmin", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.AdminID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Kullanıcı
Admin</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.AdminUserName,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Şifresi</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.AdminPassword,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Admin Rolü</label>

                        <div class="col-sm-3">
                            <select name="AdminRole" class="form-control m-b">
                                <option value="A">A</option>
                                <option value="B">B</option>
                                <option value="C">C</option>
                            </select>
                        </div>
                    </div>
                    <div class="form-group row">
                        <div class="col-sm-4 col-sm-offset-2">
                            <button class="btn btn-primary btn-sm" type="submit">Admin
Güncelle</button>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>
@model EntityLayer.Concrete.Blog
@{
    ViewData["Title"] = "UpdateBlog";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

```



```

<div class="row">
  <div class="col-lg-12">
    <div class="ibox">
      <div class="ibox-title">
        <h5>Blog Güncelleme </h5>
      </div>
      <div class="ibox-content">
        @using (Html.BeginForm("UpdateBlog", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
        {
          @Html.HiddenFor(x => x.BlogID, new { @class = "form-control",
@style = "margin-top:8px;" })
          <div class="form-group row">
            <label class="col-sm-2 col-form-label">Blog Başlığı</label>

            <div class="col-sm-3">@Html.TextBoxFor(x => x.BlogTitle, new {
@class = "form-control", @style = "" })</div>
          </div>
          <div class="hr-line-dashed"></div>
          <div class="form-group row">
            <label class="col-sm-2 col-form-label">Blog Alanı</label>

            <div class="col-sm-3">@Html.TextBoxFor(x => x.BlogContent, new
{ @class = "form-control", @style = "" })</div>
          </div>
          <div class="hr-line-dashed"></div>
          <div class="form-group row">
            <label class="col-sm-2 col-form-label">Resim</label>

            <div class="col-sm-3">
              <br />
              @Html.TextBoxFor(x => x.BlogImage, new { @class = "form-
control", @style = "" , @readonly = "readonly" })</div>
            </div>
          <div class="hr-line-dashed"></div>
          <div class="form-group row">
            <label class="col-sm-2 col-form-label">Blog Tarihi</label>

            <div class="col-sm-3">@Html.TextBoxFor(x => x.BlogCreateDate,
new { @class = "form-control", @style = "" })</div>
          </div>
          <div class="hr-line-dashed"></div>
          <div class="form-group row">
            <label class="col-sm-2 col-form-label">Yazar</label>

            <div class="col-sm-3">@Html.DropDownListFor(x => x.WriterID,
(List<SelectListItem>)ViewBag.blog, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
          </div>
          <div class="hr-line-dashed"></div>
          <div class="form-group row">
            <div class="col-sm-4 col-sm-offset-2">
              <button class="btn btn-primary btn-sm" type="submit">Blog
Güncelle</button>
            </div>
          </div>
        }
      </div>
    </div>
  </div>
</div>

```

```

</div>
@model EntityLayer.Concrete.League
@{
    ViewData["Title"] = "UpdateLeague";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox">
            <div class="ibox-title">
                <h5>lig Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdateLeague", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.LeagueID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Lig Adı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.LeagueName, new
{ @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Değeri</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.LeagueValue, new
{ @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Takım Sayısı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.LeagueTeamCount,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Kuruluş Yılı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x =>
x.LeagueFoundationYear, new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Açıklama</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x =>
x.LeagueDescription, new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <div class="col-sm-4 col-sm-offset-2">
                            <button class="btn btn-primary btn-sm" type="submit">Lig
Güncelle</button>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
@model EntityLayer.Concrete.Player
@{
    ViewData["Title"] = "UpdatePlayer";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Futbolcu Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdatePlayer", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.PlayerID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Adı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerName, new
{ @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Soyadı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerSurname,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Yaş</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerAge, new {
@class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Boy</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerHeight,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Kilo</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerWeight,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Pozisyon ISO</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x =>
x.PlayerPositionISO, new { @class = "form-control", @style = "" })</div>
                    </div>
                }
            </div>
        </div>
    </div>

```

```

<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Pozisyon</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerPosition,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Açıklama</label>

    <div class="col-sm-3">@Html.TextBoxFor(x =>
x.PlayerDescription, new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Kulüp</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerClub, new
{ @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Doğum Tarihi</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerBirthDate,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Menajer</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerManager,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Değer</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerValue, new
{ @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Numara</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.PlayerNumber,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Ülke</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Takım</label>

```

```

        <div class="col-sm-3">@Html.DropDownListFor(x => x.TeamID,
(List<SelectListItem>)ViewBag.team, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Lig</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.LeagueID,
(List<SelectListItem>)ViewBag.league, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Yetenek</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.SkillID,
(List<SelectListItem>)ViewBag.skill, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>

```

```

        <div class="form-group row">
            <div class="col-sm-4 col-sm-offset-2">
                <button class="btn btn-primary btn-sm"
type="submit">Oyuncuyu Güncelle</button>
            </div>
        </div>
    }
</div>
</div>
</div>

```

@model EntityLayer.Concrete.Skill

```

@{
    ViewData["Title"] = "UpdateSkill";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

```

```

<div class="row">
    <div class="col-lg-12">
        <div class="ibox">
            <div class="ibox-title">
                <h5>Yetenek Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdateSkill", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.SkillID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Gol</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.Goal, new {
@class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Asist</label>

```

```

        <div class="col-sm-3">@Html.TextBoxFor(x => x.Assist, new {
@class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">BBC</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.BBC, new {
@class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">AnahtarPas</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.KeyPass, new {
@class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">İsabetli Şut</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.ShootsOnTarget,
new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">İsabetsiz Şut</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.ShootsOffTarget,
new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Kırmızı Kart</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.RedCards, new {
@class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Sarı Kart</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.YellowCards, new
{ @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Başarısız
Dripling</label>

        <div class="col-sm-3">@Html.TextBoxFor(x =>
x.UnsuccessfulDribbling, new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Başarılı
Dripling</label>

        <div class="col-sm-3">@Html.TextBoxFor(x =>
x.SuccesfulDribbling, new { @class = "form-control", @style = "" })</div>
    </div>

```

```

<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Kazanılan İkili
Mücadele</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.AerialDuelsWon,
new { @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Kaybedilen İkili
Mücadele</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.AerialDuelsLost,
new { @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Pozisyon Kaybı</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.PossessionLost,
new { @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Kazanılan
Pozisyon</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.PossessionWon,
new { @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Penaltı</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.Penalties, new {
@class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Uzun Top</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.LongBall, new {
@class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">Yedek Maç
Sayısı</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.BenchMatch, new
{ @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
  <label class="col-sm-2 col-form-label">İlk 11</label>

  <div class="col-sm-3">@Html.TextBoxFor(x => x.StartedMatch,
new { @class = "form-control", @style = "" })</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">

```

```

        <label class="col-sm-2 col-form-label">Reyting</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.Rating, new {
@class = "form-control", @style = "" })</div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label">MOTM</label>

            <div class="col-sm-3">@Html.TextBoxFor(x => x.MOTM, new {
@class = "form-control", @style = "" })</div>
            </div>
            <div class="hr-line-dashed"></div>
            <div class="form-group row">
                <label class="col-sm-2 col-form-label">Yetenek Futbolcu
Adı</label>

                <div class="col-sm-3">@Html.TextBoxFor(x => x.SkillPlayerName,
new { @class = "form-control", @style = "" })</div>
                </div>
                <div class="hr-line-dashed"></div>
                <div class="form-group row">
                    <div class="col-sm-4 col-sm-offset-2">
                        <button class="btn btn-primary btn-sm"
type="submit">Yetenek Güncelle</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
@model EntityLayer.Concrete.Team
@{
    ViewData["Title"] = "UpdateTeam";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Takım Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdateTeam", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.TeamID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Takım Adı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamName, new {
@class = "form-control", @style = "" })</div>
                        </div>
                        <div class="hr-line-dashed"></div>
                        <div class="form-group row">
                            <label class="col-sm-2 col-form-label">Teknik Direktör</label>

                            <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamCoach, new {
@class = "form-control", @style = "" })</div>

```



```

</div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Takım Değeri</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamValue, new {
@class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Stat Adı</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamStatName,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Futbolcu Sayısı</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamPlayerCount,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Açıklama</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamDescription,
new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Kuruluş Yılı</label>

    <div class="col-sm-3">@Html.TextBoxFor(x =>
x.TeamFoundationYear, new { @class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Logo</label>

    <div class="col-sm-3">@Html.TextBoxFor(x => x.TeamImage, new {
@class = "form-control", @style = "" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Ülke</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
<div class="hr-line-dashed"></div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Lig</label>

    <div class="col-sm-3">@Html.DropDownListFor(x => x.LeagueID,
(List<SelectListItem>)ViewBag.league, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
<div class="hr-line-dashed"></div>

<div class="form-group row">

```

```

        <div class="col-sm-4 col-sm-offset-2">
            <button class="btn btn-primary btn-sm" type="submit">Takım
Güncelle</button>
        </div>
    </div>
}
</div>
</div>
</div>
</div>
@model EntityLayer.Concrete.Writer
@{
    ViewData["Title"] = "UpdateWriter";
    Layout = "~/Views/Shared/AdminLayout.cshtml";
}

<div class="row">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Yazar Güncelleme </h5>
            </div>
            <div class="ibox-content">
                @using (Html.BeginForm("UpdateWriter", "Admin", FormMethod.Post, new {
@class = "", enctype = "multipart/form-data" }))
                {
                    @Html.HiddenFor(x => x.WriterID, new { @class = "form-control",
@style = "margin-top:8px;" })
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Kullanıcı Adı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterUserName,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Adı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterName, new
{ @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Soyadı</label>

                        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterSurname,
new { @class = "form-control", @style = "" })</div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Resim</label>

                        <div class="col-sm-3">
                            <br />
                            @Html.TextBoxFor(x => x.WriterImage, new { @class = "form-
control", @style = "", @readonly = "readonly" })
                        </div>
                    </div>
                    <div class="hr-line-dashed"></div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Mail</label>

```

```

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterMail, new
{ @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">$ifre</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterPassword,
new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Hakkında</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterAbout, new
{ @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Ünvan</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterTitle, new
{ @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Doğum Tarihi</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterBirthDate,
new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Adres</label>

        <div class="col-sm-3">@Html.TextBoxFor(x => x.WriterAddress,
new { @class = "form-control", @style = "" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Ülke</label>

        <div class="col-sm-3">@Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-control m-b", @style =
"margin-top:8px;" })</div>
    </div>
    <div class="hr-line-dashed"></div>
    <div class="form-group row">
        <div class="col-sm-4 col-sm-offset-2">
            <button class="btn btn-primary btn-sm" type="submit">Yazar
Güncelle</button>
        </div>
    </div>
    </div>
}
</div>
</div>
</div>
</div> @using EntityLayer.Concrete
@model List<Writer>
@{
    ViewData["Title"] = "WriterIndex";
}

```

```

Layout = "~/Views/Shared/AdminLayout.cshtml";
}
<style>
    td > div {
        overflow-y: scroll;
        overflow-x: hidden;
    }
</style>
<div class="row" style="margin-top:50px; margin-left:25px;">
    <div class="col-lg-12">
        <div class="ibox ">
            <div class="ibox-title">
                <h5>Writers CRUD</h5>
            </div>
            <div class="ibox-content">

                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Kullanıcı Adı</th>
                            <th>Adı</th>
                            <th>Soyadı</th>
                            <th>Küçük Resim</th>
                            <th>Büyük Resim</th>
                            <th>Mail</th>
                            <th>Şifre</th>
                            <th>Hakkında</th>
                            <th>Ünvan</th>
                            <th>Doğum Tarihi</th>
                            <th>Kayıt Tarihi</th>
                            <th>Adres</th>
                            <th>Durum</th>
                            <th>Ülke</th>
                            <th>Güncelle</th>
                            <th>Sil</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model)
                        {
                            <tr>
                                <td>@item.WriterID</td>
                                <td>@item.WriterUserName</td>
                                <td>@item.WriterName</td>
                                <td>@item.WriterSurname</td>
                                <td></td>
                                <td><a href="/WriterImageFiles/@item.WriterImage">Resimi
Gör</a></td>
                                <td>@item.WriterMail</td>
                                <td>@item.WriterPassword</td>
                                <td>@item.WriterAbout</td>
                                <td>@item.WriterTitle</td>

                                <td>@(((DateTime)item.WriterBirthDate).ToString("dd.MM.yyyy"))</td>
                                <td>@(((DateTime)item.WriterCreateDate).ToString("dd.MM.yyyy"))</td>
                                <td height="20" style="overflow-
x:auto">@item.WriterAddress</td>
                                <td>@item.WriterStatus</td>
                                <td>@item.Country.CountryName</td>

```

```

                <td><a class="btn btn-primary"
href="/Admin/UpdateWriter/@item.WriterID/"><span class="fa fa-check"></span></a></td>
                <td><a class="btn btn-danger"
href="/Admin/DeleteWriter/@item.WriterID/"><span class="fa fa-trash"></span></a></td>
            </tr>
        }
    </tbody>
</table>
    <a class="btn btn-info" style="margin-top:10px;" asp-
action="AddWriter" asp-controller="Admin">Yazar Ekle</a>
</div>
</div>
</div>
</div>

```

Views : Blog

```

@using EntityLayer.Concrete
@model List<Blog>
@{
    ViewData["Title"] = "BlogReadAll";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<section class="section" style="margin-top:20px;">
    <div class="row">

        <div class="col-lg-3">

        </div>

        <div class="col-lg-6">
            @foreach (var item in Model)
            {
                <div class="">
                    <div class="card-header">
                        <a href="#" class="ri-account-circle-fill bx-pull-left ri-
2x"></a>
                        <a href="#" class="ql-size-small">@item.Writer.WriterName
@item.Writer.WriterSurname</a> <br />
                        <span class="text-muted small pt-2 ps-
1">@(((DateTime)item.BlogCreateDate).ToString("dd.MM.yyyy"))</span>
                    </div>

                    <div class="card mb-3">
                        <div class="row g-0">
                            <div class="col-md-4">
                                
                            </div>
                            <div class="col-md-8">
                                <div class="card-body">
                                    <h5 class="card-title">@item.BlogTitle</h5>
                                    <p class="card-text">@item.BlogContent </p>
                                    <br />
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            }
        </div>
    </div>
</div>

```

```

    }

    @await Component.InvokeAsync("CommentListByBlog", new {id = ViewBag.i})
    @await Html.PartialAsync("/Views/Comment/CommentListByBlog.cshtml")

</div>

<div class="col-lg-3">

    </div>
</div>
</section>

@using EntityLayer.Concrete
@model List<Blog>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<section class="section" style="margin-top:20px;">
    <div class="row">

        <div class="col-lg-3">

            <div class="card">
                <div class="card-body">
                    <h5 class="card-title">En Pahalı Futbolcular</h5>
                    <p>...</p>
                </div>
            </div>
            <div class="card">
                <div class="card-body">
                    <h5 class="card-title">En Çok Asist Yapanlar</h5>
                    <p>...</p>
                </div>
            </div>

        </div>

        <div class="col-lg-6">
            @foreach (var item in Model)
            {
                <div class="">
                    <div class="card-header">
                        <a href="#" class="ri-account-circle-fill bx-pull-left ri-
2x"></a>
                        <a href="/WriterProfile/WriterProfiles/@item.Writer.WriterID/"
class="ql-size-small">@item.Writer.WriterName @item.Writer.WriterSurname</a> <br />
                        <span class="text-muted small pt-2 ps-
1">@((DateTime)item.BlogCreateDate).ToString("dd.MM.yyyy")</span>
                    </div>

                    <div class="card mb-3">
                        <div class="row g-0">
                            <div class="col-md-4">
                                
                            </div>
                            <div class="col-md-8">
                                <div class="card-body">

```

```

<h5 class="card-title">@item.BlogTitle</h5>
<p class="card-text">@item.BlogContent <a
href="/Blog/BlogReadAll/@item.BlogID/" class="font-weight-bold">Devamını Oku
...</a></p>

<br />

</div>
</div>
</div>
</div>
</div>
}

</div>
<div class="col-lg-3">

    <div class="" style="height:400px;">
        <div class="">
            <div id="fs-standings"></div>
            <script>(function (w, d, s, o, f, js, fjs) { w['fsStandingsEmbed']
= o; w[o] = w[o] || function () { (w[o].q = w[o].q || []).push(arguments) }; js =
d.createElement(s), fjs = d.getElementsByTagName(s)[0]; js.id = o; js.src = f;
js.async = 1; fjs.parentNode.insertBefore(js, fjs); }(window, document, 'script',
'mw', 'https://cdn.footystats.org/embeds/standings.js')); mw('params', { leagueID:
6125 });</script>
            </div>
        </div>
        @await Component.InvokeAsync("GetMostValuablePlayer")

    </div>
</div>
</section>

```

Views : Comment

```

<textarea placeholder="Yorum gir..." class="form-control" style="height:
100px"></textarea>
<button type="button" class="btn btn-sm btn-primary bx-pull-right" style="margin-
top:10px;">
    Yorumu Paylaş
</button>

```

Views : Contacts

```

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<section class="section contact">

    <div class="row gy-4">

        <div class="col-xl-6">

            <div class="row">
                <div class="col-lg-6">

```

```

        <div class="info-box card">
            <i class="bi bi-geo-alt"></i>
            <h3>Adres</h3>
            <p>Karşıyaka/İZMİR<br>ScoutingSoftware, 35000</p>
        </div>
    </div>
    <div class="col-lg-6">
        <div class="info-box card">
            <i class="bi bi-telephone"></i>
            <h3>Bizi Ara</h3>
            <p>+90 539 583 75 17<br>...</p>
        </div>
    </div>
    <div class="col-lg-6">
        <div class="info-box card">
            <i class="bi bi-envelope"></i>
            <h3>E-mail adresimiz</h3>
            <p>info@scoutingnetwork.com<br>selcukhancirgin@gmail.com</p>
        </div>
    </div>
    <div class="col-lg-6">
        <div class="info-box card">
            <i class="bi bi-clock"></i>
            <h3>Hizmet Aralıklarımız</h3>
            <p>Pazartesi - Cuma<br>10.00 - 19.00</p>
        </div>
    </div>
</div>

<div class="col-xl-6">
    <div class="card p-4">
        <form method="post">
            <div class="row gy-4">

                <div class="col-md-6">
                    <input type="text" name="ContactUserName" class="form-
control" placeholder="Adın">
                </div>

                <div class="col-md-6">
                    <input type="email" class="form-control"
name="ContactMail" placeholder="Email Adresin">
                </div>

                <div class="col-md-12">
                    <input type="text" class="form-control"
name="ContactSubject" placeholder="Konu">
                </div>

                <div class="col-md-12">
                    <textarea class="form-control" name="ContactMessage"
rows="6" placeholder="Mesaj" ></textarea>
                </div>

                <div class="col-md-12 text-center">
                    <button class="btn btn-primary w-100"
type="submit">Gönder</button>
                </div>

            </div>
        </form>
    </div>
</div>

```



```

        </form>
    </div>

</div>

</div>

</section>

```

Views : ErrorPage

```

@{
    Layout = null;
}

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1.0" name="viewport">

    <title>Pages / 404</title>
    <meta content="" name="description">
    <meta content="" name="keywords">

    <!-- Favicons -->
    <link href="assets/img/favicon.png" rel="icon">
    <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

    <!-- Google Fonts -->
    <link href="https://fonts.gstatic.com" rel="preconnect">
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

    <!-- Vendor CSS Files -->
    <link href="~/NiceAdmin/assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/boxicons/css/boxicons.min.css"
rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/quill/quill.snow.css" rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/quill/quill.bubble.css" rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/remixicon/remixicon.css" rel="stylesheet">
    <link href="~/NiceAdmin/assets/vendor/simple-datatables/style.css"
rel="stylesheet">

    <!-- Template Main CSS File -->
    <link href="~/NiceAdmin/assets/css/style.css" rel="stylesheet">

    <!-- =====
    * Template Name: NiceAdmin - v2.2.2
    * Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-
template/
    * Author: BootstrapMade.com
    * License: https://bootstrapmade.com/license/
    ===== -->
</head>

```

```

<body>

    <main>
        <div class="container">

            <section class="section error-404 min-vh-100 d-flex flex-column align-items-center justify-content-center">
                <h1>404</h1>
                <h2>Aradığınız sayfa bulunamadı.</h2>
                <a class="btn" href="index.html">Anasayfaya dön</a>
                
                <div class="credits">
                    <!-- All the links in the footer should remain intact. -->
                    <!-- You can delete the links only if you purchased the pro version. -->
                    <!-- Licensing information: https://bootstrapmade.com/license/ -->
                    <!-- Purchase the pro version with working PHP/AJAX contact form: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/ -->
                </div>
            </section>

        </div>
    </main><!-- End #main -->

    <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

    <!-- Vendor JS Files -->
    <script src="~/NiceAdmin/assets/vendor/apexcharts/apexcharts.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/chart.js/chart.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/echarts/echarts.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/quill/quill.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/simple-datatables/simple-datatables.js"></script>
    <script src="~/NiceAdmin/assets/vendor/tinymce/tinymce.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/php-email-form/validate.js"></script>

    <!-- Template Main JS File -->
    <script src="~/NiceAdmin/assets/js/main.js"></script>

</body>

</html>

```

Views : League

```

@using EntityLayer.Concrete
@model List<League>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

@foreach (var item in Model)
{
    <div class="col-xxl-3 col-md-6">
        <div class="card info-card sales-card">

```

```

        <div class="filter">
            <a class="icon" href="#" data-bs-toggle="dropdown"><i class="bi bi-
three-dots"></i></a>
            <ul class="dropdown-menu dropdown-menu-end dropdown-menu-arrow">
                <li class="dropdown-header text-start">
                    <h6>Seenekler</h6>
                </li>
                <li><a class="dropdown-item"
href="/Player/PlayerDetails/">Detaylar</a></li>
            </ul>
        </div>

        <div class="card-body">
            <h5 class="card-title">Kuruluş Yılı : @item.LeagueFoundationYear
<span>| Takım sayısı : @item.LeagueTeamCount</span></h5>

            <div class="d-flex align-items-center">
                <div class="card-icon rounded-circle d-flex align-items-center
justify-content-center">
                    <i class="ri-indeterminate-circle-fill"></i>
                </div>
                <div class="ps-3">
                    <h6>@item.LeagueName</h6>
                    <span class="text-success small pt-1 fw-bold">Değer :
@item.LeagueValue</span> <span class="text-muted small pt-2 ps-1"></span>
                </div>
            </div>
        </div>
    </div>
</div>
}

```

Views : Login

```

@model EntityLayer.Concrete.Writer
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/RegisterLoginLayout.cshtml";
}

<div class="card col-6">

    <div class="card-body">

        <div class="pt-4 pb-2">
            <h5 class="card-title text-center pb-0 fs-4">Giriş Yap</h5>
        </div>

        @using (Html.BeginForm("Index", "Login", FormMethod.Post, new { @class = "row
g-3" }))
        {
            <div class="col-12">
                <label for="yourName" class="form-label">Kullanıcı Adı</label>
                <input type="text" name="WriterUserName" class="form-control" id="">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterUserName)</p>
            </div>
            <div class="col-12">
                <label for="yourName" class="form-label">Şifre</label>
                <input type="password" name="WriterPassword" class="form-control"
id="">
            </div>
        }
    </div>
</div>

```

```

                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterPassword)</p>
            </div>
            <div class="col-12">
                <button class="btn btn-primary w-100" type="submit">Giriş Yap</button>
            </div>
            <div class="col-12">
                <p class="small mb-0">Hesabın yok mu? <a asp-action="Index" asp-
controller="Register">Hesap Oluştur</a></p>
            </div>
        }

    </div>
</div>

```

Views : Player

```

@using EntityLayer.Concrete
@model List<Player>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}
@foreach (var item in Model)
{
    <div class="col-xxl-3 col-md-6">
        <div class="card info-card sales-card">

            <div class="filter">
                <a class="icon" href="#" data-bs-toggle="dropdown"><i class="bi bi-three-
dots"></i></a>
                <ul class="dropdown-menu dropdown-menu-end dropdown-menu-arrow">
                    <li class="dropdown-header text-start">
                        <h6>Seçenekler</h6>
                    </li>
                    <li><a class="dropdown-item"
href="/Player/PlayerDetails/@item.PlayerID/">Detaylar</a></li>
                </ul>
            </div>

            <div class="card-body">
                <h5 class="card-title">@item.PlayerNumber <span>|
@item.Team.TeamName</span></h5>

                <div class="d-flex align-items-center">
                    <div class="card-icon rounded-circle d-flex align-items-center
justify-content-center">
                        <i class="ri-account-circle-fill"></i>
                    </div>
                    <div class="ps-3">
                        <h6>@item.PlayerName @item.PlayerSurname</h6>
                        <span class="text-success small pt-1 fw-
bold">@item.PlayerPositionISO</span> <span class="text-muted small pt-2 ps-
1">@item.Country.CountryName</span>
                    </div>
                </div>
            </div>
        </div>
    </div>
}

```

```

@using EntityLayer.Concrete
@model List<Player>
@{
    ViewData["Title"] = "PlayerDetails";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<section class="section" style="margin-top:20px;">
    <div class="row">
        <div class="col-lg-12">
            <div class="card" style="height:1200px">
                <div class="card-body">

                    <h5 class="card-title">@ViewBag.pn @ViewBag.ps Detayları</h5><br

                />

                @foreach (var item in Model)
                {
                    <div class="container bootstrap snippets bootdey"
style="margin-right:225px;">
                        <div class="panel-body inf-content">
                            <div class="row">
                                <div class="col-md-4">
                                    
                                </div>
                                <div class="col-md-6">
                                    <strong></strong><br>
                                    <div class="table-responsive">
                                        <table class="table table-user-
information">
                                            <tbody>
                                                <tr>
                                                    <td>
                                                        <strong>
                                                            <span class="glyphicon
glyphicon-asterisk text-primary"></span>
                                                            Forma Numarası
                                                        </strong>
                                                    </td>
                                                    <td class="text-primary">
                                                        @item.PlayerNumber
                                                    </td>
                                                </tr>
                                                <tr>
                                                    <td>
                                                        <strong>
                                                            <span class="glyphicon
glyphicon-user text-primary"></span>
                                                            Adı
                                                        </strong>
                                                    </td>
                                                    <td class="text-primary">
                                                        @item.PlayerName
                                                    </td>
                                                </tr>
                                                <tr>
                                                    <td>

```

glyphicon-user text-primary">	
	Soyadı
	
	</td>
	<td class="text-primary">
	@item.PlayerSurname
	</td>
	</tr>
	<tr>
	<td>
glyphicon-user text-primary">	
	Takımı
	
	</td>
	<td class="text-primary">
	@item.Team.TeamName
	</td>
	</tr>
	<tr>
	<td>
glyphicon-user text-primary">	
	Pozisyon
	
	</td>
	<td class="text-primary">
	@item.PlayerPosition
	</td>
	</tr>
	<tr>
	<td>
glyphicon-user text-primary">	
	Değeri
	
	</td>
	<td class="text-primary">
	@item.PlayerValue
	</td>
	</tr>
	<tr>
	<td>
glyphicon-user text-primary">	
	Ülkesi
	
	</td>
<td class="text-primary">	@item.Country.CountryName
	</td>
	</tr>
	<tr>

```

        <td>
            <strong>
                <span class="glyphicon glyphicon-user text-primary"></span>
                Doğum Tarihi
            </strong>
        </td>
        <td class="text-primary">
            @((DateTime)item.PlayerBirthDate).ToString("dd.MM.yyyy")
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
<br />
<div class="card col-lg-4" style="float:left; margin-right:20px; margin-left:40px;">
    <div class="card-body pb-0">
        <h5 class="card-title">Genel Özellikler</h5>

        <div id="trafficChart" style="min-height: 400px;"
class="echart"></div>

        <script>
            document.addEventListener("DOMContentLoaded", ()
=> {
echarts.init(document.querySelector("#trafficChart")).setOption({
    tooltip: {
        trigger: 'item'
    },
    legend: {
        top: '5%',
        left: 'center'
    },
    series: [{
        name: '@item.PlayerName',
        type: 'pie',
        radius: ['40%', '70%'],
        avoidLabelOverlap: false,
        label: {
            show: false,
            position: 'center'
        },
        emphasis: {
            label: {
                show: true,
                fontSize: '18',
                fontWeight: 'bold'
            }
        },
        labelLine: {
            show: false
        },
        data: [{
            value: @item.Skill.MOTM,

```

```

        name: 'MOTM'
    },
    {
        value: @item.Skill.Rating,
        name: 'Rating'
    },
    {
        value: @item.Skill.Penalties,
        name: 'Penalt1'
    },
    {
        value: @item.Skill.Goal,
        name: 'Gol'
    },
    {
        value: @item.Skill.Assist,
        name: 'Asist'
    }
]
    });
});
</script>

</div>
</div>

<div class="card col-lg-4">
    <div class="card-body pb-0">
        <h5 class="card-title">Genel İstatistikler</h5>

        <div id="budgetChart" style="min-height: 400px;"
class="echart"></div>

        <script>
            document.addEventListener("DOMContentLoaded", ()
=> {
                var budgetChart =
echarts.init(document.querySelector("#budgetChart")).setOption({
                    legend: {
                        data: ['Kazanılan', 'Kaybedilen']
                    },
                    radar: {
                        // shape: 'circle',
                        indicator: [{
                            name: 'İkili Mücadele',
                            max: @item.Skill.AerialDuelsWon +
@item.Skill.AerialDuelsLost
                        },
                        {
                            name: 'Pozisyon',
                            max: @item.Skill.PossesionWon +
@item.Skill.PossesionLost
                        },
                        {
                            name: 'Dribbling',
                            max:
@item.Skill.SuccesfulDribbling + @item.Skill.UnsuccessfulDribbling
                        },
                        {
                            name: 'Maç Sayısı',

```



```

        @using (Html.BeginForm("Index", "Register", FormMethod.Post, new { @class =
"row g-3", enctype = "multipart/form-data" } ) )
        {
            <div class="col-6">
                <label for="yourName" class="form-label">Ad</label>
                <input type="text" name="WriterName" class="form-control" id="">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterName)</p>
            </div>
            <div class="col-6">
                <label for="yourName" class="form-label">Soyad</label>
                <input type="text" name="WriterSurname" class="form-control" id="">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterSurname)</p>
            </div>

            <div class="col-12">
                <label for="yourEmail" class="form-label">Kullanıcı Adı</label>
                <input type="text" name="WriterUserName" class="form-control" id="">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterUserName)</p>
            </div>

            <div class="col-12">
                <label for="yourEmail" class="form-label">Email</label>
                <input type="text" name="WriterMail" class="form-control"
id="yourEmail">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterMail)</p>
            </div>
            <div class="col-12">
                <label for="yourEmail" class="form-label">Şifre</label>
                <input type="password" name="WriterPassword" class="form-control"
id="">
                <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterPassword)</p>
            </div>
            <div class="col-12">
                <label for="yourEmail" class="form-label">Profil Resmi</label>
                <input class="form-control" name="WriterImage" type="file">
                <p class="text-danger">@*@Html.ValidationMessageFor(x =>
x.WriterImage)*@</p>
            </div>

            <div class="col-6">
                <label for="inputDate" class="form-label">Doğum Tarihi</label>
                <div class="">
                    <input type="datetime" name="WriterBirthDate" class="form-
control">
                    <p class="text-danger">@Html.ValidationMessageFor(x =>
x.WriterBirthDate)</p>
                </div>
            </div>
            <div class="col-6">
                @Html.Label("Ülke")
                @Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-select", @style = "margin-
top:8px;" })
            </div>
            <div class="col-12">
                <button class="btn btn-primary w-100" type="submit" >Hesap
Oluştur</button>

```

```

        </div>
        <div class="col-12">
            <p class="small mb-0">Zaten hesabın var mı? <a asp-action="Index" asp-
controller="Login">Giriş yap</a></p>
        </div>
    }
</div>
</div>

```

Views : Shared : Components

```

@model IEnumerable<EntityLayer.Concrete.Comment>
<div class="row" style="padding:10px;">
    <h2 class="border-bottom">Yorumlar</h2>
</div>
@foreach (var item in Model)
{
    <div class="card mb-4">
        <div class="card-body" style="padding:10px;">
            <p>@item.CommentContent</p>

            <div class="d-flex justify-content-between">
                <div class="d-flex flex-row align-items-center">
                    <span class="ri-account-circle-fill"></span>
                    <p class="small mb-0 ms-2">@item.CommentUserName</p>
                </div>
                <div class="d-flex flex-row align-items-center">
                    </div>
            </div>
        </div>
    </div>
</div>
}

@using EntityLayer.Concrete
@model List<Player>

<div class="card info-card sales-card">

    <div class="card-body">
        <h5 class="card-title">En Değerli 3 Futbolcu </h5>

        @foreach (var item in Model)
        {

            <h5 class="card-title">@item.PlayerNumber <span>|
@item.Team.TeamName</span></h5>

            <div class="d-flex align-items-center">
                <div class="card-icon rounded-circle d-flex align-items-center
justify-content-center">
                    <i class="ri-account-circle-fill"></i>
                </div>
                <div class="ps-3">
                    <h6>@item.PlayerName @item.PlayerSurname</h6>
                    <span class="text-success small pt-1 fw-
bold">@item.PlayerPositionISO</span> <span class="text-muted small pt-2 ps-
1">@item.Country.CountryName</span>
                </div>
            </div>
        }
    </div>

```

```

    }

</div>

</div>
@using EntityLayer.Concrete
@model List<Blog>
<section class="section" style="margin-top:20px;">
    <div class="row">
        <div class="col-lg-3">

            <div class="card">
                <section class="section profile">
                    <div class="card-body profile-card pt-4 d-flex flex-column
align-items-center">

                        <h2>@ViewBag.un @ViewBag.sn</h2>
                        <h3 class="">@ViewBag.tt</h3>
                        <div class="social-links mt-2">

                        </div>
                    </div>
                </section>
            </div>

        </div>

        <div class="col-lg-6">
            @foreach (var item in Model)
            {
                <div class="">
                    <div class="card-header">
                        <a href="#" class="ri-account-circle-fill bx-pull-left ri-
2x"></a>
                        <a href="#" class="ql-size-small">@ViewBag.un
@ViewBag.sn</a> <br />
                        <span class="text-muted small pt-2 ps-
1">@(((DateTime)item.BlogCreateDate).ToString("dd.MM.yyyy"))</span>
                    </div>

                    <div class="card mb-3">
                        <div class="row g-0">
                            <div class="col-md-4">
                                
                            </div>
                            <div class="col-md-8">
                                <div class="card-body">
                                    <h5 class="card-title">@item.BlogTitle</h5>
                                    <p class="card-text">@item.BlogContent <a
href="/Blog/BlogReadAll/@item.BlogID/" class="font-weight-bold">Devamını Oku
...</a></p>
                                <br />
                            </div>
                        </div>
                    </div>
                </div>
            }
        </div>
    </div>

```

```

        </div>
    </div>
}

</div>

<div class="col-lg-3">
    <div class="" style="height:400px;">
        <div class="">
            <div id="fs-standings"></div>
            <script>(function (w, d, s, o, f, js, fjs) {
w['fsStandingsEmbed'] = o; w[o] = w[o] || function () { (w[o].q = w[o].q ||
[]).push(arguments) }; js = d.createElement(s), fjs = d.getElementsByTagName(s)[0];
js.id = o; js.src = f; js.async = 1; fjs.parentNode.insertBefore(js, fjs); }(window,
document, 'script', 'mw', 'https://cdn.footystats.org/embeds/standings.js'));
mw('params', { leagueID: 6125 });</script>
            </div>
        </div>
        @await Component.InvokeAsync("GetMostValuablePlayer")
    </div>

</div>
</section>

```

```

@using EntityLayer.Concrete
@model List<Blog>
<div class="row">
    <div class="col-lg-3"></div>
    <button type="button" class="col-lg-6 btn btn-lg btn-primary" style="margin-
left:15px;" data-bs-toggle="modal" data-bs-target="#largeModal">
        Gönderi Oluştur
        <span class="ri-add-box-line bx-pull-right"></span>
    </button>
    <div class="col-lg-3"></div>
</div>

<div class="modal fade" id="largeModal" tabindex="-1">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Gönderi Oluştur</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="card-body">

                    <form class="row g-3 needs-validation" method="post"
enctype="multipart/form-data">

                        <div class="col-12">
                            <label for="yourName" class="form-
label">Başlık</label>

```

```

        <input type="text" name="BlogTitle" class="form-
control" id="">
        <div class="invalid-feedback">Please, enter your
name!</div>
    </div>
    <div class="col-12">
        <label for="yourName" class="form-
label">İçerik</label>
        <textarea placeholder="Ne Düşünüyorsun ?"
name="BlogContent" class="form-control" style="height: 100px"></textarea>
        <div class="invalid-feedback">Please, enter your
name!</div>
    </div>
    <div class="col-12">
        <input class="form-control" name="BlogImage"
type="file">
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-
bs-dismiss="modal">Vazgeç</button>
        <button type="submit" class="btn btn-
primary">Paylaş</button>
    </div>
</form>
</div>
</div>
</div>
</div>
<br />
<br />
<section class="section" style="margin-top:20px;">
    <div class="row">
        <div class="col-lg-3">
            <div class="card">
                <section class="section profile">
                    <div class="card-body profile-card pt-4 d-flex flex-column
align-items-center">
                        
                        <h2>@ViewBag.un</h2>
                        <h3 class="">@ViewBag.tt</h3>
                        <div class="social-links mt-2">
                            </div>
                    </div>
                </section>
            </div>
        </div>
        <div class="col-lg-6">
            @foreach (var item in Model)
            {
                <div class="">

```

```

        <div class="card-header">
            <a href="#" class="ri-account-circle-fill bx-pull-left ri-
2x"></a>
            <a href="#" class="ql-size-small">@ViewBag.n
@ViewBag.s</a> <br />
            <span class="text-muted small pt-2 ps-
1">@((DateTime)item.BlogCreateDate).ToString("dd.MM.yyyy")</span>
        </div>

        <div class="card mb-3">
            <div class="row g-0">
                <div class="col-md-4">
                    
                </div>
                <div class="col-md-8">
                    <div class="card-body">
                        <h5 class="card-title">@item.BlogTitle</h5>
                        <p class="card-text">@item.BlogContent <a
href="/Blog/BlogReadAll/@item.BlogID/" class="font-weight-bold">Devamını Oku
...</a></p>
                    <br />
                </div>
            </div>
        </div>
    </div>
}

</div>

<div class="col-lg-3">

    <div class="" style="height:400px;">
        <div class="">
            <div id="fs-standings"></div>
            <script>(function (w, d, s, o, f, js, fjs) {
w['fsStandingsEmbed'] = o; w[o] = w[o] || function () { (w[o].q = w[o].q ||
[]).push(arguments) }; js = d.createElement(s), fjs = d.getElementsByTagName(s)[0];
js.id = o; js.src = f; js.async = 1; fjs.parentNode.insertBefore(js, fjs); }(window,
document, 'script', 'mw', 'https://cdn.footystats.org/embeds/standings.js'));
mw('params', { leagueID: 6125 });</script>
            </div>
        </div>
        @await Component.InvokeAsync("GetMostValuablePlayer")

    </div>

</div>
</section>

```

Views : Shared

```
@{
    Layout = null;
}

<!DOCTYPE html>
<html>

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin Paneli</title>

    <link href="~/inspinia-gh-pages/css/bootstrap.min.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/font-awesome/css/font-awesome.css"
rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/plugins/iCheck/custom.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/animate.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/style.css" rel="stylesheet">

</head>

<body>

    <div id="wrapper">

        <nav class="navbar-default navbar-static-side" role="navigation">
            <div class="sidebar-collapse">
                <ul class="nav metismenu" id="side-menu">
                    <li>
                        <a asp-action="Index" asp-controller="Admin"><i class="fa fa-
bars"></i> <span class="nav-label">Futbolcular</span></a>
                    </li>
                    <li>
                        <a asp-action="SkillIndex" asp-controller="Admin"><i class="fa
fa-bars"></i> <span class="nav-label">Yetenekler</span></a>
                    </li>
                    <li>
                        <a asp-action="TeamIndex" asp-controller="Admin"><i class="fa
fa-bars"></i> <span class="nav-label">Takımlar</span></a>
                    </li>
                    <li>
                        <a asp-action="LeagueIndex" asp-controller="Admin"><i
class="fa fa-bars"></i> <span class="nav-label">Ligler</span></a>
                    </li>
                    <li>
                        <a asp-action="BlogIndex" asp-controller="Admin"><i class="fa
fa-bars"></i> <span class="nav-label">Bloglar</span></a>
                    </li>
                    <li>
                        <a asp-action="WriterIndex" asp-controller="Admin"><i
class="fa fa-bars"></i> <span class="nav-label">Yazarlar</span></a>
                    </li>
                    <li>
                        <a asp-action="CommentIndex" asp-controller="Admin"><i
class="fa fa-bars"></i> <span class="nav-label">Yorumlar</span></a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>
</body>
</html>
```



```

        <a asp-action="AdminIndex" asp-controller="Admin"><i class="fa
fa-bars"></i> <span class="nav-label">Adminler</span></a>
        </li>
        <li>
            <a asp-action="ContactIndex" asp-controller="Admin"><i
class="fa fa-bars"></i> <span class="nav-label">Gelen Kutusu</span></a>
            </li>
            <li>
                <a asp-action="CountryIndex" asp-controller="Admin"><i
class="fa fa-bars"></i> <span class="nav-label">Ülkeler</span></a>
                </li>
                <li>
                    <a asp-action="Logout" asp-controller="Admin"><i class="fa fa-
sign-out"></i> <span class="nav-label">Çıkış Yap</span></a>
                    </li>
            </ul>

        </div>
    </nav>

    <div id="page-wrapper" class="gray-bg">
        <div class="row border-bottom">
            <nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
                <div class="navbar-header">
                    <a class="navbar-minimalize minimize-styl-2 btn btn-primary
" href="#"><i class="fa fa-bars"></i> </a>

                </div>

            </nav>
        </div>

        @RenderBody()

    </div>
</div>
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<link href="https://cdn.bootcss.com/sweetalert/1.1.3/sweetalert.min.css"
rel="stylesheet">

<script>
    function confirmation(ev, itemname) {

        ev.preventDefault();
        var urlToRedirect = ev.currentTarget.getAttribute('href'); //use
currentTarget because the click may be on the nested i tag and not a tag causing the
href to be empty
        console.log(urlToRedirect); // verify if this is the right URL
        swal({
            title: "Silmek istediğine emin misin?",
            text: itemname + " adlı veri silinecek ",
            icon: "warning",
            buttons: ["Vazgeç", "Sil"],
            dangerMode: true,
        })
        .then((willDelete) => {
            if (willDelete) {

                window.location = urlToRedirect;

```

```

        swal("İşlem başarılı!", {
            icon: "success",
            buttons:false
        });
    } else {
        swal("Veri silinemedi.");
    }
});
}
function login() {
    swal({
        title: "Tebrikler!",
        text: "Kayıt Başarılı!",
        icon: "success",
        button: false,
    });
}
</script>

<!-- Mainly scripts -->
<script src="~/inspinia-gh-pages/js/jquery-3.1.1.min.js"></script>
<script src="~/inspinia-gh-pages/js/popper.min.js"></script>
<script src="~/inspinia-gh-pages/js/bootstrap.js"></script>
<script src="~/inspinia-gh-
pages/js/plugins/metisMenu/jquery.metisMenu.js"></script>
<script src="~/inspinia-gh-
pages/js/plugins/slimscroll/jquery.slimscroll.min.js"></script>

<!-- Peity -->
<script src="~/inspinia-gh-pages/js/plugins/peity/jquery.peity.min.js"></script>

<!-- Custom and plugin javascript -->
<script src="~/inspinia-gh-pages/js/inspinia.js"></script>
<script src="~/inspinia-gh-pages/js/plugins/pace/pace.min.js"></script>

<!-- iCheck -->
<script src="~/inspinia-gh-pages/js/plugins/iCheck/ichack.min.js"></script>

<!-- Peity -->
<script src="~/inspinia-gh-pages/js/demo/peity-demo.js"></script>

<script>
    $(document).ready(function () {
        $('i-checks').iCheck({
            checkboxClass: 'icheckbox_square-green',
            radioClass: 'iradio_square-green',
        });
    });</script>
</body>
</html>

@{
    Layout = null;
}

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
  <link href="https://cdn.bootcss.com/sweetalert/1.1.3/sweetalert.min.css"
rel="stylesheet">

  <script>
    function confirmation(ev, itemname) {

      ev.preventDefault();
      var urlToRedirect = ev.currentTarget.getAttribute('href'); //use
currentTarget because the click may be on the nested i tag and not a tag causing the
href to be empty
      console.log(urlToRedirect); // verify if this is the right URL
      swal({
        title: "Silmek istediğine emin misin?",
        text: itemname + "adlı oyuncu silinecek ",
        icon: "warning",
        buttons: ["Vazgeç", "Sil"],
        dangerMode: true,
      })
      .then((willDelete) => {
        if (willDelete) {

          window.location = urlToRedirect;
          swal("İşlem başarılı!", {
            icon: "success",
          });
        } else {
          swal("Veri silinemedi.");
        }
      });
    }
    function login() {

      swal({
        title: "Tebrikler!",
        text: "Kayıt Başarılı!",
        icon: "success",
        button: false,
      });
    }
  </script>

  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Pages / Register - NiceAdmin Bootstrap Template</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="~/NiceAdmin/assets/img/favicon.png" rel="icon">
  <link href="~/NiceAdmin/assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.gstatic.com" rel="preconnect">
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700
,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,60
0,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->

```

```

    <link href=~/{NiceAdmin/assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/boxicons/css/boxicons.min.css"
rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/quill/quill.snow.css" rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/quill/quill.bubble.css" rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/remixicon/remixicon.css" rel="stylesheet">
    <link href=~/{NiceAdmin/assets/vendor/simple-datatables/style.css"
rel="stylesheet">

    <!-- Template Main CSS File -->
    <link href=~/{NiceAdmin/assets/css/style.css" rel="stylesheet">

    <!-- =====
    * Template Name: NiceAdmin - v2.2.2
    * Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-
template/
    * Author: BootstrapMade.com
    * License: https://bootstrapmade.com/license/
    ===== -->
</head>

<body>

    <main>
        <div class="container">

            <section class="section register min-vh-100 d-flex flex-column align-
items-center justify-content-center py-4">
                <div class="container">
                    <div class="row justify-content-center">
                        <div class="col-lg-6 col-md-6 d-flex flex-column align-items-
center justify-content-center">

                            <div class="d-flex justify-content-center py-4">
                                <a href="index.html" class="logo d-flex align-items-
center w-auto">
                                    <img src=~/{NiceAdmin/assets/img/logo.png" alt="">
                                    <span class="d-none d-lg-
block">ScoutingNetwork</span>
                                </a>
                            </div><!-- End Logo -->

                            @RenderBody()

                            <div class="credits">
                                </div>

                        </div>
                    </div>
                </div>
            </section>

        </div>
    </main><!-- End #main -->

    <a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>

```

```

    <!-- Vendor JS Files -->
    <script src="~/NiceAdmin/assets/vendor/apexcharts/apexcharts.min.js"></script>
    <script
src="~/NiceAdmin/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/chart.js/chart.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/echarts/echarts.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/quill/quill.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/simple-datatables/simple-
datatables.js"></script>
    <script src="~/NiceAdmin/assets/vendor/tinymce/tinymce.min.js"></script>
    <script src="~/NiceAdmin/assets/vendor/php-email-form/validate.js"></script>

    <!-- Template Main JS File -->
    <script src="~/NiceAdmin/assets/js/main.js"></script>

</body>

</html>

@{
    Layout = null;
}

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1.0" name="viewport">

    <title>Dashboard - NiceAdmin Bootstrap Template</title>
    <meta content="" name="description">
    <meta content="" name="keywords">

    <link href="~/inspinia-gh-pages/css/bootstrap.min.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/font-awesome/css/font-awesome.css"
rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/plugins/slick/slick.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/plugins/slick/slick-theme.css"
rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/animate.css" rel="stylesheet">
    <link href="~/inspinia-gh-pages/css/style.css" rel="stylesheet">
    <style>
        .slick_demo_2 .ibox-content {
            margin: 0 10px;
        }
    </style>

    <script src="~/inspinia-gh-pages/js/plugins/slick/slick.min.js"></script>

    <!-- Favicons -->
    <link href="~/NiceAdmin/assets/img/favicon.png" rel="icon">
    <link href="~/NiceAdmin/assets/img/apple-touch-icon.png" rel="apple-touch-icon">

    <!-- Google Fonts -->
    <link href="https://fonts.gstatic.com" rel="preconnect">
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700
,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,60
0,600i,700,700i" rel="stylesheet">

```

```

<!-- Vendor CSS Files -->
<link href=~\NiceAdmin\assets\vendor\bootstrap\css\bootstrap.min.css"
rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\bootstrap-icons\bootstrap-icons.css"
rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\boxicons\css\boxicons.min.css"
rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\quill\quill.snow.css" rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\quill\quill.bubble.css" rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\remixicon\remixicon.css" rel="stylesheet">
<link href=~\NiceAdmin\assets\vendor\simple-datatables\style.css"
rel="stylesheet">

<!-- Template Main CSS File -->
<link href=~\NiceAdmin\assets\css\style.css" rel="stylesheet">

<!-- =====
* Template Name: NiceAdmin - v2.2.2
* Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-
template/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
===== -->
</head>

<body>

<!-- ===== Header ===== -->
<header id="header" class="header fixed-top d-flex align-items-center">

    <div class="d-flex align-items-center justify-content-between">
        <a href="index.html" class="logo d-flex align-items-center">
            <img src=~\NiceAdmin\assets\img\logo.png" alt="">
            <span class="d-none d-lg-block">ScoutingNetwork</span>
        </a>
        <i class="bi bi-list toggle-sidebar-btn"></i>
    </div><!-- End Logo -->

    <div class="search-bar">
        <form class="search-form d-flex align-items-center" method="POST"
action="#">
            <input type="text" name="query" placeholder="Ara" title="...">
            <button type="submit" title="Search"><i class="bi bi-
search"></i></button>
        </form>
    </div><!-- End Search Bar -->

    <nav class="header-nav ms-auto">
        <ul class="d-flex align-items-center">

            <li class="nav-item d-block d-lg-none">
                <a class="nav-link nav-icon search-bar-toggle " href="#">
                    <i class="bi bi-search"></i>
                </a>
            </li><!-- End Search Icon-->

        </ul>
    </nav><!-- End Icons Navigation -->

</header><!-- End Header -->
<!-- ===== Sidebar ===== -->
<aside id="sidebar" class="sidebar">

```

```

<ul class="sidebar-nav" id="sidebar-nav">

    <li class="nav-item">
        <a class="nav-link " asp-action="index" asp-controller="Blog">
            <i class="bi bi-grid"></i>
            <span>Anasayfa</span>
        </a>
    </li><!-- End Dashboard Nav -->

    <li class="nav-item">
        <a class="nav-link collapsed" data-bs-target="#player-nav" data-bs-
toggle="collapse" href="#">
            <i class="bi bi-menu-button-wide"></i><span>Futbolcular</span><i
class="bi bi-chevron-down ms-auto"></i>
        </a>
        <ul id="player-nav" class="nav-content collapse " data-bs-
parent="#sidebar-nav">
            <li>
                <a asp-action="index" asp-controller="Player">
                    <i class="bi bi-circle"></i><span>Tüm futbolcular</span>
                </a>
            </li>
        </ul>
    </li>
    <li class="nav-item">
        <a class="nav-link collapsed" data-bs-target="#team-nav" data-bs-
toggle="collapse" href="#">
            <i class="bi bi-menu-button-wide"></i><span>Takımlar</span><i
class="bi bi-chevron-down ms-auto"></i>
        </a>
        <ul id="team-nav" class="nav-content collapse " data-bs-
parent="#sidebar-nav">
            <li>
                <a asp-action="index" asp-controller="Player">
                    <i class="bi bi-circle"></i><span>Kulüp Takımları</span>
                </a>
            </li>
        </ul>
    </li>
    <li class="nav-item">
        <a class="nav-link collapsed" data-bs-target="#league-nav" data-bs-
toggle="collapse" href="#">
            <i class="bi bi-menu-button-wide"></i><span>Ligler</span><i
class="bi bi-chevron-down ms-auto"></i>
        </a>
        <ul id="league-nav" class="nav-content collapse " data-bs-
parent="#sidebar-nav">
            <li>
                <a asp-action="index" asp-controller="League">
                    <i class="bi bi-circle"></i><span>Tüm Ligler</span>
                </a>
            </li>
        </ul>
    </li>

    <li class="nav-heading">-</li>

    <li class="nav-item">
        <a class="nav-link collapsed" asp-action="index" asp-
controller="WriterProfile">

```

```

                <i class="bi bi-person"></i>
                <span>Profilim</span>
            </a>
        </li><!-- End Profile Page Nav -->

        <li class="nav-item">
            <a class="nav-link collapsed" asp-action="index" asp-controller="SSS">
                <i class="bi bi-question-circle"></i>
                <span>S.S.S</span>
            </a>
        </li><!-- End F.A.Q Page Nav -->

        <li class="nav-item">
            <a class="nav-link collapsed" asp-action="index" asp-
controller="Contact">
                <i class="bi bi-envelope"></i>
                <span>Destek</span>
            </a>
        </li><!-- End Contact Page Nav -->
        <!-- End Register Page Nav -->
        <li class="nav-item">
            <a class="nav-link collapsed" asp-action="WriterSettings" asp-
controller="WriterProfile">
                <i class="bx bx-cog"></i>
                <span>Ayarlar</span>
            </a>
        </li>
        <li class="nav-item">
            <a class="nav-link collapsed" asp-action="Logout" asp-
controller="Login">
                <i class="bi bi-box-arrow-in-right"></i>
                <span>Çıkış</span>
            </a>
        </li><!-- End Login Page Nav -->

    </ul>

</aside><!-- End Sidebar-->

<main id="main" class="main">

    <section class="section dashboard">
        <div class="row">

            <div class="col-lg-12">
                <div class="row">
                    @RenderBody()
                    <!-- Footballer Card -->
                </div>
            </div>
        </div>
    </section>

</main><!-- End #main -->
<!-- ===== Footer ===== -->
<!-- End Footer -->

    <a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>

    <!-- Vendor JS Files -->
    <script src="~/NiceAdmin/assets/vendor/apexcharts/apexcharts.min.js"></script>

```



```

<script
src=~\NiceAdmin\assets\vendor\bootstrap\js\bootstrap.bundle.min.js"></script>
<script src=~\NiceAdmin\assets\vendor\chart.js\chart.min.js"></script>
<script src=~\NiceAdmin\assets\vendor\echarts\echarts.min.js"></script>
<script src=~\NiceAdmin\assets\vendor\quill\quill.min.js"></script>
<script src=~\NiceAdmin\assets\vendor\simple-datatables\simple-
datatables.js"></script>
<script src=~\NiceAdmin\assets\vendor\tinymce\tinymce.min.js"></script>
<script src=~\NiceAdmin\assets\vendor\php-email-form\validate.js"></script>

<!-- Template Main JS File -->
<script src=~\NiceAdmin\assets\js\main.js"></script>

</body>

</html>

```

Views : SSS

```

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<h1>S.S.S</h1>

```

Views : Team

```

@using EntityLayer.Concrete
@model List<Team>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}
@foreach (var item in Model)
{
    <div class="col-xxl-3 col-md-6">
        <div class="card info-card sales-card">

            <div class="filter">
                <a class="icon" href="#" data-bs-toggle="dropdown"><i class="bi bi-
three-dots"></i></a>
                <ul class="dropdown-menu dropdown-menu-end dropdown-menu-arrow">
                    <li class="dropdown-header text-start">
                        <h6>Seenekler</h6>
                    </li>
                    <li><a class="dropdown-item"
href="/Player/PlayerDetails/">Detaylar</a></li>
                </ul>
            </div>

            <div class="card-body">
                <h5 class="card-title">Kuruluş : @item.TeamFoundationYear <span>|
Futbolcu sayısı : @item.TeamPlayerCount</span></h5>

                <div class="d-flex align-items-center">
                    <div class="card-icon rounded-circle d-flex align-items-center
justify-content-center">
                        <i class="ri-shield-fill"></i>
                    </div>

```

```

                <div class="ps-3">
                    <h6>@item.TeamName</h6>
                    <span class="text-success small pt-1 fw-
bold">@item.Country.CountryISO</span> <span class="text-muted small pt-2 ps-
1">@item.Country.CountryName</span>
                </div>
            </div>
        </div>
    </div>
</div>
}

```

Views : WriterProfile

```

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

@await Component.InvokeAsync("WriterProfileLastBlog", new { id = ViewBag.i })

@{
    ViewData["Title"] = "WriterProfiles";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

@await Component.InvokeAsync("WriterLastBlog", new { id = ViewBag.i })

@model EntityLayer.Concrete.Writer
@{
    ViewData["Title"] = "WriterSettings";
    Layout = "~/Views/Shared/UserLeftLayout.cshtml";
}

<section class="section profile">
    <div class="row">
        <div class="col-xl-4">

            <div class="card">
                <div class="card-body profile-card pt-4 d-flex flex-column align-
items-center">

                    
                    <h2>@ViewBag.n @ViewBag.s</h2>
                    <h3>@ViewBag.tt</h3>
                </div>
            </div>

        </div>

        <div class="col-xl-8">

            <div class="card">
                <div class="card-body pt-3">
                    <!-- Bordered Tabs -->
                    <ul class="nav nav-tabs nav-tabs-bordered">

```

```

        <li class="nav-item">
            <button class="nav-link active" data-bs-toggle="tab" data-
bs-target="#profile-overview">Genel</button>
        </li>

        <li class="nav-item">
            <button class="nav-link" data-bs-toggle="tab" data-bs-
target="#profile-edit">Profili Düzenle</button>
        </li>

    </ul>
    <div class="tab-content pt-2">

        <div class="tab-pane fade show active profile-overview"
id="profile-overview">
            <h5 class="card-title">Hakkında</h5>
            <p class="small fst-italic">@ViewBag.ab</p>

            <h5 class="card-title">Profil Detayları</h5>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-heading">Adı
Soyadı</label>
                <div class="col-lg-9 col-md-8">@ViewBag.n
@ViewBag.s</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-
heading">Kullanıcı Adı</label>
                <div class="col-lg-9 col-md-8">@ViewBag.un</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-
heading">Ünvan</label>
                <div class="col-lg-9 col-md-8">@ViewBag.tt</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-
heading">Ülke</label>
                <div class="col-lg-9 col-md-8">@ViewBag.ct</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-
heading">Adres</label>
                <div class="col-lg-9 col-md-8">@ViewBag.ad</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-heading">Doğum
Tarihi</label>
                <div class="col-lg-9 col-md-8">@ViewBag.dt</div>
            </div>

            <div class="row">
                <label class="col-md-4 col-lg-3 alert-heading">E-Mail
adresi</label>
                <div class="col-lg-9 col-md-8">@ViewBag.ml</div>
            </div>
        </div>
    </div>

```

```

</div>

<div class="tab-pane fade profile-edit pt-3" id="profile-
edit">

    <!-- Profile Edit Form -->
    @using (Html.BeginForm("WriterSettings", "WriterProfile",
FormMethod.Post, new { @class = "row g-3", enctype = "multipart/form-data" }))
    {
        @Html.HiddenFor(x => x.WriterID, new { @class = "form-
control", @style = "margin-top:8px;" })
        <div class="row mb-3">
            <label for="profileImage" class="col-md-4 col-lg-3
col-form-label">Profil Resmi</label>
            <div class="col-md-8 col-lg-9">
                
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Adı</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterName" type="text"
class="form-control" value="@ViewBag.n">
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Soyadı</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterSurname" type="text"
class="form-control" value="@ViewBag.s">
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Kullanıcı Adı</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterUserName" type="text"
class="form-control" value="@ViewBag.un">
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Hakkında</label>
            <div class="col-md-8 col-lg-9">
                <textarea name="WriterAbout" class="form-
control" style="height: 100px">@ViewBag.ab</textarea>
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Ünvan</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterTitle" type="text"
class="form-control" value="@ViewBag.tt">
            </div>
        </div>
    }
}

```

```

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-label">E-
mail</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterMail" type="text"
class="form-control" value="@ViewBag.ml">
            </div>
        </div>

        <div class="row mb-3">
            <label for="Country" class="col-md-4 col-lg-3 col-
form-label">Ülke</label>
            <div class="col-md-8 col-lg-9">
                @Html.DropDownListFor(x => x.CountryID,
(List<SelectListItem>)ViewBag.country, new { @class = "form-select", @style = "margin-
top:8px;" })
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Adres</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterAddress" type="text"
class="form-control" value="@ViewBag.ad">
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Doğum Tarihi</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterBirthDate" type="datetime"
class="form-control" value="05.05.2022">
            </div>
        </div>

        <div class="row mb-3">
            <label class="col-md-4 col-lg-3 col-form-
label">Şifre</label>
            <div class="col-md-8 col-lg-9">
                <input name="WriterPassword" type="password"
readonly="readonly" class="form-control" value="@ViewBag.pass">
            </div>
        </div>

        <div class="text-center">
            <button type="submit" class="btn btn-
primary">Değişiklikleri Kaydet</button>
        </div>
    }<!-- End Profile Edit Form -->

</div>

</div><!-- End Bordered Tabs -->

</div>

```

```

        </div>
    </div>
</section>

```

Startup.cs

```

using FluentValidation.AspNetCore;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc.Authorization;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SocialScoutingNetwork
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to
        the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllers(x => x.AllowEmptyInputInBodyModelBinding = true);
            services.AddControllersWithViews();
            services.AddSession();
            services.AddAuthorization(options =>
            {
                options.AddPolicy("Admin",
                    authBuilder =>
                    {
                        authBuilder.RequireRole("Administrators");
                    });
            });
            services.AddMvc().AddFluentValidation(fv =>
            {
                fv.ConfigureClientsideValidation(enabled: false);
            });
            services.AddMvc(config => {
                var policy = new AuthorizationPolicyBuilder()
                    .RequireAuthenticatedUser()
                    .Build();
                config.Filters.Add(new AuthorizeFilter(policy));
            });
        }
    }
}

```

```

    });
    services.AddMvc();
    services.AddAuthentication(
        CookieAuthenticationDefaults.AuthenticationScheme)
        .AddCookie(x => {
            x.LoginPath = "/Login/Index";
        });
    services.AddAuthentication(
        CookieAuthenticationDefaults.AuthenticationScheme)
        .AddCookie("Admin", x => {
            x.LoginPath = "/Admin/AdminLoginIndex";
        });
}

// This method gets called by the runtime. Use this method to configure the
HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for
production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }

    app.UseStatusCodePagesWithReExecute("/ErrorPage/Error1", "?code={0}");
    app.UseHttpsRedirection();
    app.UseStaticFiles();
    app.UseSession();
    app.UseRouting();
    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
}
}

```

TARTIŞMA VE SONUÇ

Site başlamadan önce istediğim genel özelliklere genel olarak sahip bir site oldu. Bir sosyal ağda bulunması gereken genel özelliklere neredeyse sahip. Admin kısmında ise yazarların kullandığı siteye neredeyse yüzde yüz etki edebiliyorum. Projenin geliştirilmesi şüphesiz ki hiç bitmeyecektir. Özellikle site içerisinde bulunan tasarımlar ve genel kod yapısı yıllarledikçe eskiyecek olacağından ve kullanıcıların gözünde sıradanlaşacağı için yenilik ve geliştirilme olasılığı çok fazla gözüküyor.

KAYNAKLAR

1. www.stackoverflow.com
2. <https://sweetalert.js.org/>
3. www.microsoft.com
4. Murat Yücedağ Youtube kanalı
5. Bootstrap

EKLER