|  |  |
| --- | --- |
|  | **İZMİR BAKIRÇAY ÜNİVERSİTESİ**  **MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**  **BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ** |

**<GeziLog>**

# ****Sistem Gereksinimleri ve Özellikleri Raporu****

2021-2022 GÜZ

BİL 203 & BİL 209 Dönem Projesi

Sürüm 3.0

*Hazırlayan*

*200601010\_Kübra\_Akkaya*

*200601044\_Şevval\_Özdamar*

İçindekiler

İçindekiler 2

Revizyon Geçmişi 2

1. Giriş 3

1.1 Amaç 3

1.2 Ürün Kapsamı 3

1.3 Kurallar, Tanımlar ve Kısaltmalar 3

1.4 Referanslar 3

1.5 Rapora Genel Bakış 3

2. Genel Tanım 4

2.1 Ürüne Bakış 4

2.2 Ürün İşlevleri 4

2.3 Kullanıcı Sınıfları ve Özellikleri 4

2.4 Çalışma Ortamı 4

2.5 Tasarım ve Uygulama Kısıtlamaları 5

2.6 Kullanıcı Belgeleri 5

2.7 Varsayımlar ve Bağımlılıklar 5

2.8 Riskler 5

3. Harici Arayüz Gereksinimleri 6

3.1 Kullanıcı Arayüzleri 6

3.2 Donanım Arayüzleri 6

3.3 Yazılım Arayüzleri 6

3.4 İletişim Arayüzleri 6

4. Sistem Özellikleri 7

4.1 <Sistem Özelliği 1> 7

4.2 <Sistem Özelliği 2> 7

5. Yazılım 8

6. Sistem Tasarımı 9

6.1 Varlık-İlişki Diyagramı 9

6.2 İlişkisel Cebir İfadeleri 9

6.3 Sınıf Diyagramı 9

6.4 *CRUD* Matrisi 9

6.5 View, Trigger ve Stored Procedure Kullanımı 9

Ek A: SQL İfadeleri 10

Ek B: Ekran Görüntüleri 11

Ek C: To Be Determined List (TBDL) 12

Revizyon Geçmişi

|  |  |  |  |
| --- | --- | --- | --- |
| **Adı** | **Tarihi** | **Değişiklik Nedeni** | **Sürüm** |
| Taslak | 20.11.2021 | Sistem gereksinimleri ve özellikleri şablonunun oluşturulması | 1.0 |
| Faz 1 | 30.11.2021 | Sisteme ait ilk gereksinimlerin ve açıklamaların rapora eklenmesi | 1.1 |
| Faz 2 | 26.12.2021 | Sistemin geliştiği noktaların somut verilerle rapora eklenmesi | 2.0 |
| Final | 02.02.2022 | Sistemin tüm eksiklerinin giderilmesi ve rapora kaydedilmesi. | 3.0 |

## Giriş

Bu bölümde önerilen sistem için tanımlayıcı bilgilere yer verilir.

Sistem bir seyahat rehberi sitesidir. Blog yazarları (adminler); bloglarda gezdiği yerlere, mekanlara yer verir ve önerilerde bulunur. Bu blogları, site ziyaretçileri siteye üye olmadan okuyabilmektedir. Seyahat tavsiyeleri almak isteyenler için yararlı bir site olması amaçlanmaktadır.

### Amaç

Bu raporda gereksinimleri belirtilen ürünü tanımlayın. İlgili faydalar, amaçlar ve hedefler dahil olmak üzere belirtilen yazılımın ve amacının kısa bir tanımını ifade edin.

Geliştiriciler, proje yöneticileri, pazarlama personeli, kullanıcılar, testçiler ve rapor yazarları gibi raporun amaçlandığı hedef okuyucu türlerini tanımlayın.

Amaç: Yazılan bloglar ile seyahat yapacak kişilerin gezi rehberi olmak. Hangi şehirde nereler gezilebilir, nerelerde konaklanabilir, hangi restoranlarda yemek yenebilir gibi sorulara cevap vermek. Ziyaretçilerin seyahatlerini daha verimli bir şekilde geçirmelerini sağlamak.

Hedef okuyucu: Seyahat ederken gezi rehberine ihtiyaç duyanlar ve blog yazıları okumayı, takip etmeyi sevenler.

### Ürün Kapsamı

Özellikle bu rapor, tüm sistemin yalnızca bir bölümünü veya tek bir alt sistemini açıklıyorsa, bu rapora söz konusu olan ürünün kapsamını açıklayın.

Yazılımı kurumsal hedeflerle veya iş stratejileriyle ilişkilendirin.

Ayrı bir vizyon ve kapsam belgesi mevcutsa, onu referans edin.

Rapor tüm sistemi kapsadığı için alt sistem yoktur.

### Kurallar, Tanımlar ve Kısaltmalar

Özel önemi olan yazı tipleri veya vurgulama gibi bu raporu yazarken izlenen standartları veya tipografik kuralları açıklayın. Örneğin, daha yüksek seviyeli gereksinimler için önceliklerin ayrıntılı gereksinimler tarafından miras alınıp alınmadığını veya her gereksinim ifadesinin kendi önceliğine sahip olup olmayacağını belirtin.

Raporu anlamaya yardımcı olacak ve alana özel terim ve kavramın tanımlarını verin ve rapor içinde kullanılan kısaltmaları alfabetik olarak listeleyin.

Db: database

### Referanslar

Bu raporun atıfta bulunduğu diğer belgeleri veya web adreslerini listeleyin. Bunlar, kullanıcı arayüzü stil kılavuzlarını, sözleşmeleri, standartları, sistem gereksinimleri spesifikasyonlarını, kullanım senaryosu belgelerini veya bir vizyon ve kapsam belgesini içerebilir. Okuyucunun başlık, yazar, sürüm numarası, tarih ve kaynak veya konum dahil olmak üzere her referansın bir kopyasına erişebilmesi için yeterli bilgiyi sağlayın.

Bu rapor şablonu için, “IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications” belgesi referans alınmıştır.

“draw.io” sitesindeki uml şablonları referans alınarak bu rapordaki uml diyagramları oluşturulmuştur.

### Rapora Genel Bakış

Bu raporun geri kalanının neler içerdiğini ve nasıl bir bölüm düzeninde olduğunu açıklayın. Raporu okumak için genel bakış bölümlerinden (ikinci bölümden) başlayarak ve her bir okuyucu tipine en uygun bölümlerden geçerek bir sıra önerin.

Raporun 2. bölümünde sistemin genel tanımları (içeriği, işlevleri, bağımlılıkları, kısıtlamaları vb.),

Raporun 3. bölümünde arayüz gereksinimleri,

Raporun 4. bölümünde sistemin özellikleri,

Raporun 5. bölümünde yazılım mimarisi,

Raporun 6. bölümünde db. içerikleri bulunmaktadır.

## Genel Tanım

Sistemin içeriğini kısaca tanıtın.

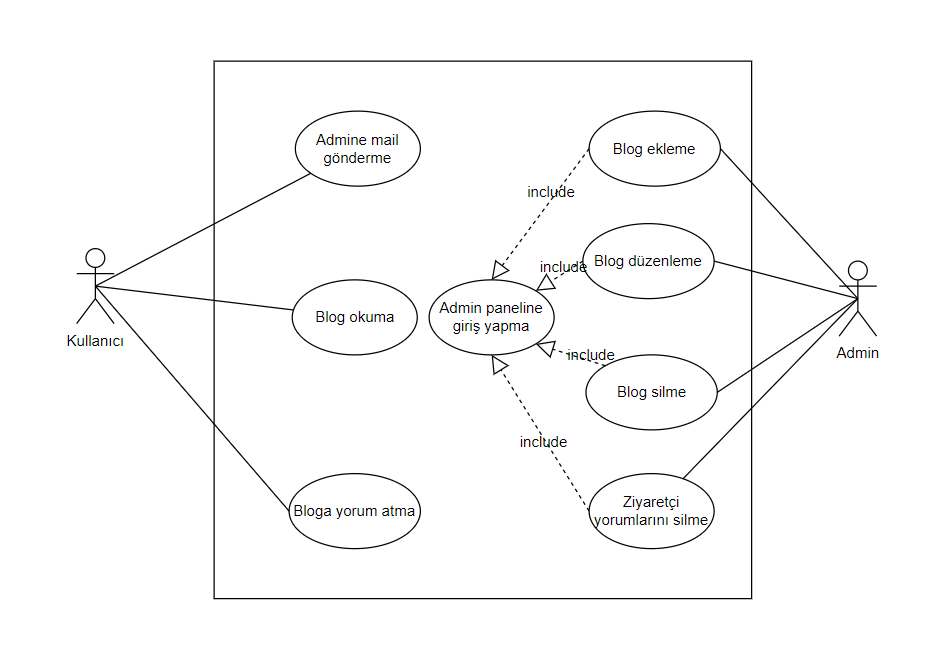
Kullanım senaryolarını ve sistem işlemlerini oluşturun. Bunlar paydaşları belirlemenize yardımcı olacaktır. Paydaşların sistemdeki rolleri nelerdir? Onları listeleyin. Data flow, use case, activity, sequence gibi UML diyagramları çizilebilir.

Bu raporda yer alan sistem tasarımı tarafından desteklenen işlevselliğin anlaşılmasına yardımcı olacaksa, bu bölümde projenin arka planını tartışabilirsiniz.

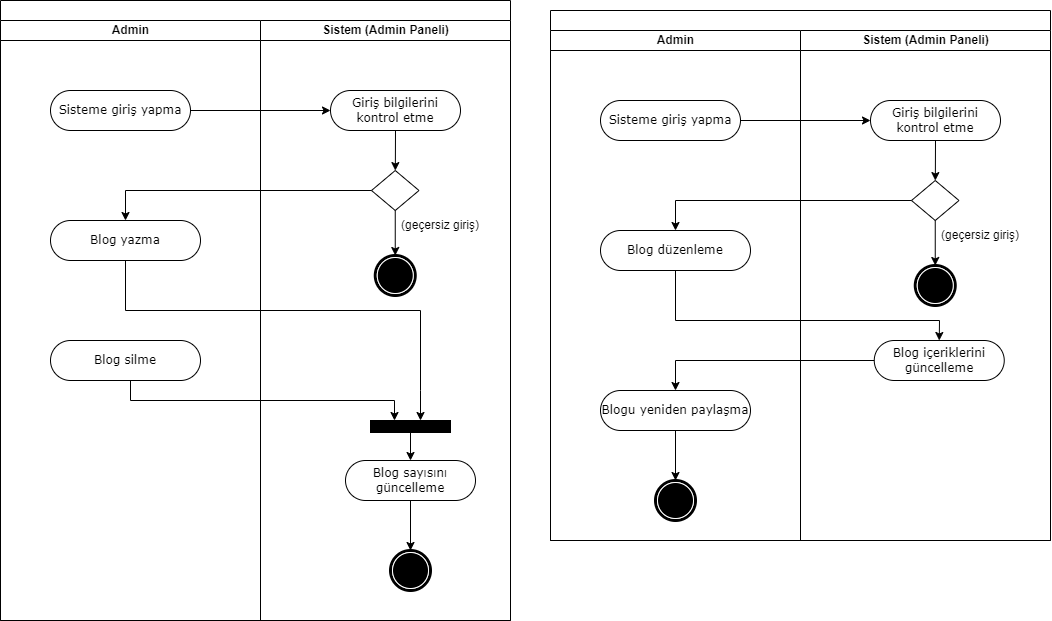
**Sistemin İçeriği:**

Şehir tanıtımları, gezilecek yerler, gidilmesi gereken mekanlar gibi konularda yazılan blogların adminler tarafından üretilip paylaşılmasıdır. Adminler, gerekli durumlarda bu blogları düzenleyebilir veya silebilir. Ziyaretçiler ise yazıları okuyabilir ve yazılara yorum atarak birbirleri ile etkileşime girebilir. Adminler, ziyaretçilerin bıraktığı yorumları kötü içerik barındırıyorsa silme hakkına sahiptir.

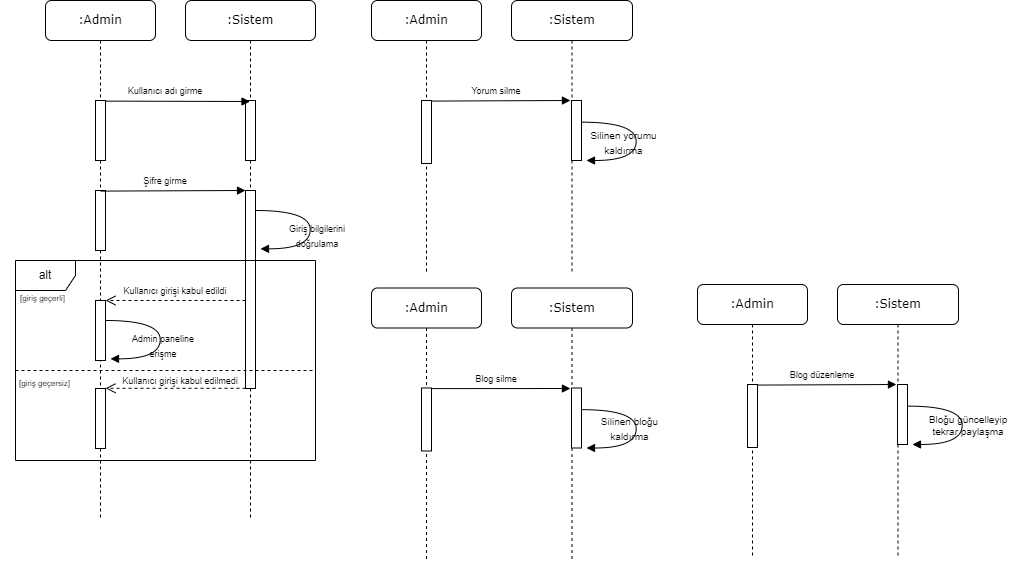
**Use Case Diyagramı:**



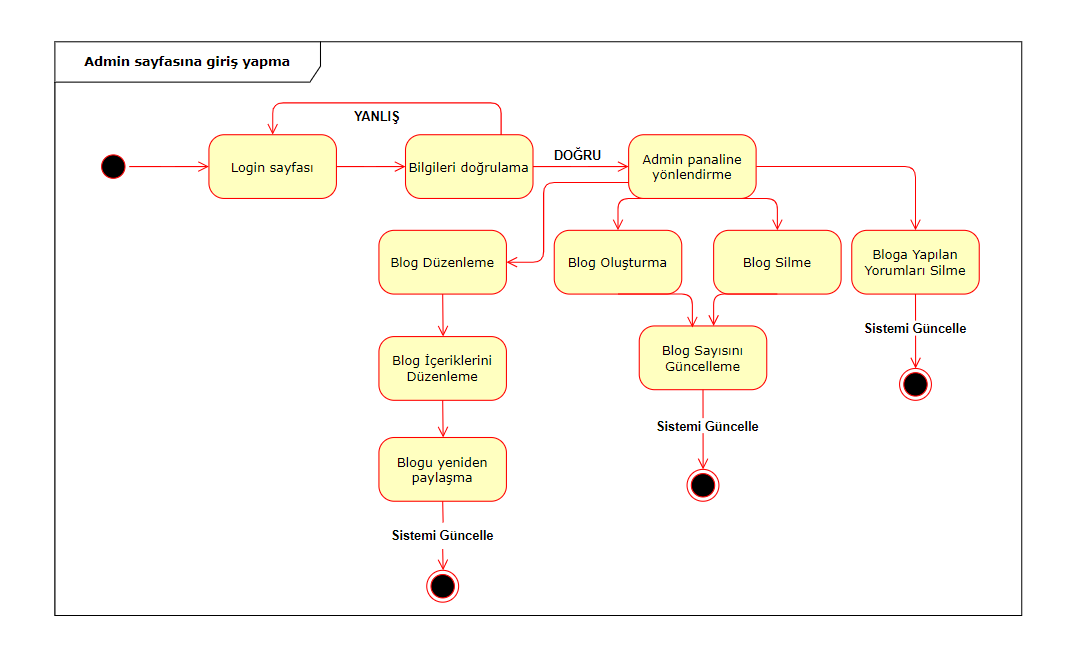
**Activity Diyagramı:**



**Sequence Diyagramı:**



**State Diyagramı:**



### Ürüne Bakış

Bu raporda belirtilen ürünün bağlamını ve menşeini tanımlayın. Örneğin, bu ürünün bir ürün ailesinin devam üyesi mi, belirli mevcut sistemlerin yedeği mi yoksa yeni, bağımsız bir ürün mü olduğunu belirtin.

Rapor daha büyük bir sistemin bir bileşenini tanımlıyorsa, daha büyük sistemin gereksinimlerini bu yazılımın işlevselliği ile ilişkilendirin ve ikisi arasındaki arayüzleri tanımlayın. Genel sistemin ana bileşenlerini, alt sistem ara bağlantılarını ve harici arabirimleri gösteren basit bir blok diyagram ya da diyagramlar yardımcı olabilir.

Bu sistem bağımsız bir üründür.

### Ürün İşlevleri

Ürünün gerçekleştirmesi gereken veya kullanıcının gerçekleştirmesine izin vermesi gereken ana işlevleri özetleyin. Ayrıntılar Bölüm 4'te verilecektir, bu nedenle burada yalnızca yüksek düzeyde bir özet (madde işareti listesi gibi) gereklidir. İşlevleri, raporun herhangi bir okuyucusu için anlaşılır kılmak için metinsel ya da grafiksel gösterimleri kullanarak düzenleyebilirsiniz. Bir üst düzey data flow diyagramı veya object class diyagramı gibi, ilgili gereksinimlerin ana gruplarının ve bunların nasıl ilişkili olduğunun bir resmi genellikle etkili olabilir. Söz konusu diyagramlar ürünün tasarımını göstermez, sadece işlevler arasındaki mantıksal ilişkiyi ifade eder.

Ziyaretçiler, bloglara yorum atabilir.

Ziyaretçiler, form üzerinden admine mesaj atabilir.

Admin, ziyaretçi tarafından atılan yorumları silebilir.

Admin; sisteme blog ekleyebilir, silebilir ve bloğu düzenleyebilir.

### Kullanıcı Sınıfları ve Özellikleri

Bu ürünü kullanacağını tahmin ettiğiniz çeşitli kullanıcı sınıflarını tanımlayın. Kullanıcı sınıfları, kullanım sıklığına, kullanılan ürün işlevlerinin alt kümesine, teknik uzmanlığa, güvenlik veya ayrıcalık seviyelerine, eğitim düzeyine veya deneyime göre farklılaştırılabilir. Her kullanıcı sınıfının ilgili özelliklerini tanımlayın. Belirli gereksinimler yalnızca belirli kullanıcı sınıflarıyla ilgili olabilir. Bu ürün için en önemli kullanıcı sınıflarını, tatmin edilmesi daha az önemli olanlardan ayırt edin.

Hedef kullanıcılar: Seyahat ederken tavsiye almak isteyenler, blog okumayı ve takip etmeyi sevenler. Kişiler bakımından bir ayrım yoktur. Her türden, her yaştan, her eğitim düzeyinden biri siteyi ziyaret edebilir.

### Çalışma Ortamı

Donanım platformu, işletim sistemi ve sürümleri ve uyum içinde bir arada var olması gereken diğer yazılım bileşenleri veya uygulamaları dahil olmak üzere yazılımın çalışacağı ortamı tanımlayın.

Bilgisayar, tablet, telefon gibi cihazlardan herhangi bir web tarayıcısı üzerinden ulaşılabilir.

### Tasarım ve Uygulama Kısıtlamaları

Geliştiricilerin kullanabileceği seçenekleri sınırlayacak öğeleri veya sorunları tanımlayın. Bunlar şunları içerebilir: kurumsal veya düzenleyici politikalar; donanım sınırlamaları (zamanlama gereksinimleri, birincil ve ikincil bellek gereksinimleri); diğer uygulamalara arayüzler; kullanılacak belirli teknolojiler, araçlar ve veri tabanları; yedekleme ve kurtarma kısıtları; paralel işlemler; dil gereksinimleri; iletişim protokolleri; Güvenlik Hususları; tasarım sözleşmeleri veya programlama standartları (örneğin, müşterinin kuruluşu teslim edilen yazılımın bakımından sorumlu olacaksa).

Blog yazılarının dili Türkçe olacağı için sistemin hitap edeceği kitle dile bağımlı olarak küçülür ve kısıtlanabilir.

Ziyaretçiler bloglara yorum atmak dışında başka etkileşimlere de girmek isteyebilirler.

### Kullanıcı Belgeleri

Yazılımla birlikte teslim edilecek kullanıcı belgeleri bileşenlerini (kullanım kılavuzları, çevrimiçi yardım ve öğreticiler gibi) listeleyin. Bilinen tüm kullanıcı belgeleri teslim biçimlerini veya standartlarını tanımlayın.

Ziyaretçiye herhangi bir tanıtım veya çevrimiçi yardım yapılmayacaktır.

### Varsayımlar ve Bağımlılıklar

Raporda belirtilen gereksinimleri etkileyebilecek varsayılan etkenleri (bilinen gerçeklerin aksine) listeleyin. Bunlar, kullanmayı planladığınız üçüncü şahıs veya ticari bileşenleri, geliştirme veya işletim ortamıyla ilgili sorunları veya kısıtlamaları içerebilir. Bu varsayımların yanlış olması, paylaşılmaması veya değişmesi durumunda proje etkilenebilir. Ayrıca, başka bir projeden yeniden kullanmayı düşündüğünüz yazılım bileşenleri gibi, başka bir yerde (örneğin, vizyon ve kapsam belgesinde veya proje planında) belgelenmemişlerse, projenin dış etkenlere olan bağımlılıklarını da belirleyin.

Sitenin güncelliği ve aktifliği adminlere bağlıdır. Çünkü siteyi ayakta tutan blog yazıları sadece adminler tarafından sisteme eklenebilmektedir.

Adminlerin atılan yorumları silmesi ve blog eklemesi, silmesi, düzenlemesi sisteme giriş yapabilmesine bağlıdır.

### Riskler

*Sistem tasarımı ve önerilen stratejilerle ilgili tüm riskleri tanımlayın.*

Sistemin tüm web tarayıcılarında çalışması ön görülmüştür fakat sürüm uyuşmazlığından kaynaklanan sorunlardan dolayı bu istek sağlanamayabilir.

Blog yazılarındaki mekanlar, konumlar ve gezi anlatımları gibi sunulan içerikler ziyaretçinin okuduğu zamana uyumluluk göstermeyebilir veya ziyaretçi sunulan içeriği beğenmeyebilir.

## Harici Arayüz Gereksinimleri

### Kullanıcı Arayüzleri

Yazılım ürünü ve kullanıcılar arasındaki her bir arabirimin mantıksal özelliklerini tanımlayın. Bu, örnek ekran görüntüleri, herhangi bir GUI standartları veya izlenecek ürün ailesi stil kılavuzları, ekran düzeni kısıtlamaları, her ekranda görünecek standart düğmeler ve işlevler (örn. yardım), klavye kısayolları, hata mesajı görüntüleme standartları ve yakın zamanda. Kullanıcı arabiriminin gerekli olduğu yazılım bileşenlerini tanımlayın. Kullanıcı arayüzü tasarımının detayları ayrı bir kullanıcı arayüzü spesifikasyonunda belgelenmelidir.

[*https://localhost:44377/Login/Login*](https://localhost:44377/Login/Login) *uzantısı ile* admin paneline gidilir. Karşımıza login ekranı çıkar. Giriş yapabilmek için admin olmak gerekir. Kullanıcı adı ve şifre ile admin paneline giriş yapabilir.

Admin sayfasına giriş yaptıktan sonra; bloglar kısmından hangi işlem yapılacaksa (silme, düzenleme ya da blog detayına gitme) o bloğa dair satırdaki butonlara basılır. Blog eklemek için, blog listesinin alt kısmında bulunan blog ekle butonuna basarak bloğa dair istenilen bilgileri girmeniz gerekir. Yorumlar için de yine aynı şekilde bloglarda söylenilen işlemler yapabilir.

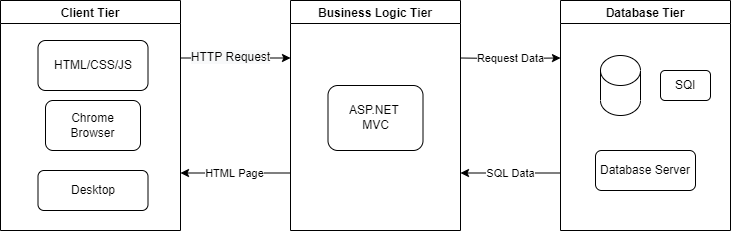
[*https://localhost:44377/Default/Index*](https://localhost:44377/Default/Index) *uzantısı ile* ana sayfaya gidilir. Bu anasayfada en son eklenen blogların ikisi gözükür. Devamını okumak için “read more” yazısına tıklanabilir. Tüm blogları görmek için, sayfanın navigation-bar bölümünden “bloglar” kısmına tıklanmalıdır. Burada yine bloğun devamını okumak için ilgili bloğun üstüne tıklanmalıdır.

Sayfanın navigation-bar bölümünden “hakkımızda” kısmından blog yazarı olan adminler hakkında bilgilere ulaşılabilir.

Yine aynı şekilde sayfanın navigation-bar bölümünden “iletişim” kısmından ilgili form doldurarak adminlerle iletişim kurulabilir.

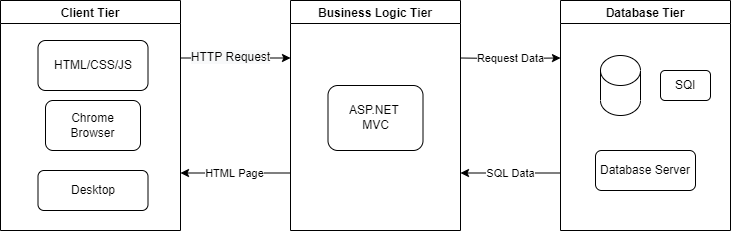
### Yazılım Arayüzleri

Veritabanları, işletim sistemleri, araçlar, kitaplıklar ve entegre ticari bileşenler dahil olmak üzere bu ürün ile diğer belirli yazılım bileşenleri (ad ve sürüm) arasındaki bağlantıları açıklayın. Sisteme giren ve çıkan veri öğelerini veya mesajları tanımlayın ve her birinin amacını açıklayın. İhtiyaç duyulan hizmetleri ve iletişimin doğasını açıklayın. Ayrıntılı uygulama programlama arabirimi protokollerini açıklayan belgelere bakın. Yazılım bileşenleri arasında paylaşılacak verileri belirleyin. Veri paylaşım mekanizmasının belirli bir şekilde uygulanması gerekiyorsa (örneğin, çok görevli bir işletim sisteminde global bir veri alanının kullanılması), bunu bir uygulama kısıtlaması olarak belirtin.



### İletişim Arayüzleri

E-posta, web tarayıcısı, ağ sunucusu iletişim protokolleri, elektronik formlar vb. dahil olmak üzere bu ürünün gerektirdiği tüm iletişim işlevleriyle ilgili gereksinimleri açıklayın. İlgili mesaj formatını tanımlayın. FTP veya HTTP gibi kullanılacak tüm iletişim standartlarını tanımlayın. Tüm iletişim güvenliği veya şifreleme sorunlarını, veri aktarım hızlarını ve senkronizasyon mekanizmalarını belirtin.



## Sistem Özellikleri

Bu şablon, ürün tarafından sağlanan başlıca hizmetler olan sistem özelliklerine göre ürün için işlevsel gereksinimlerin organize edilmesini gösterir. Bu bölümü, ürününüz için en mantıklı olanı, kullanım durumu, çalışma modu, kullanıcı sınıfı, nesne sınıfı, işlevsel hiyerarşi veya bunların kombinasyonlarına göre düzenlemeyi tercih edebilirsiniz.

### < Ziyaretçilerin bloglara yorum atabilmeleri >

Gerçekten “Sistem Özelliği 1” demeyin. Özellik adını birkaç kelimeyle belirtin.

#### Tanım ve Öncelik

Özelliğin kısa bir tanımını yapın ve Yüksek, Orta veya Düşük öncelikli olup olmadığını belirtin. Ayrıca fayda, ceza, maliyet ve risk gibi belirli öncelikli bileşen derecelendirmelerini de dahil edebilirsiniz (her biri en düşük 1 ile en yüksek 9 arasında göreceli bir ölçekte derecelendirilmiştir).

Bu özellikte sayfayı ziyaret eden ziyaretçiler isterlerse düşüncelerini ve görüşlerini blogların altına yorum olarak atabilirler. Orta önceliklidir. Yapılan yorumlar olumlu yönde olursa, bu sayfa için değerli bir niteliktir.

#### Uyaran/Yanıt Dizileri

Bu özellik için tanımlanan davranışı uyaran kullanıcı eylemlerinin ve sistem yanıtlarının sırasını listeleyin. Bunlar, kullanım durumları ile ilişkili diyalog öğelerine karşılık gelir.

Bu özelliğin uyaranı sayfayı ziyaret eden ziyaretçilerdir. Ziyaretçilerin yorumlarını yazıp gönder butonuna basmasıyla bu özellik tetiklenmiş olur. Yapılan yorum; front-end’de blog sayfasında yer alan yorumlar kısmına, back-end’de comments tablosuna kaydolur.

#### İşlevsel Gereklilikler

Bu özellikle ilişkili ayrıntılı işlevsel gereksinimleri maddeleyin. Bunlar, kullanıcının özellik tarafından sağlanan hizmetleri yerine getirmesi veya kullanım senaryosunu yürütmesi için mevcut olması gereken yazılım yetenekleridir. Ürünün beklenen hata koşullarına veya geçersiz girdilere nasıl yanıt vermesi gerektiğini dahil edin. Gereksinimler özlü, eksiksiz, açık, doğrulanabilir ve gerekli olmalıdır. Gerekli bilgilerin henüz mevcut olmadığını belirtmek için “TBD”yi (to-be-determined) kullanın. Her gereksinim, bir sıra numarası veya bir tür anlamlı etiketle (REQ-1, REQ-2, …) benzersiz bir şekilde tanımlanmalıdır.

Yorum panelinde doldurulması gereken alanları doldurmadan yorum atılırsa uyarı mesajı çıkar. O alanları doldurmadan sayfaya yorum atamazlar.

### < Adminin yapılan yorumları silebilmesi >

Gerçekten “Sistem Özelliği 1” demeyin. Özellik adını birkaç kelimeyle belirtin.

#### Tanım ve Öncelik

Özelliğin kısa bir tanımını yapın ve Yüksek, Orta veya Düşük öncelikli olup olmadığını belirtin. Ayrıca fayda, ceza, maliyet ve risk gibi belirli öncelikli bileşen derecelendirmelerini de dahil edebilirsiniz (her biri en düşük 1 ile en yüksek 9 arasında göreceli bir ölçekte derecelendirilmiştir).

Sayfaya her türlü yorum yapılabilir. Ancak üslup kuralları dışında ya da herhangi bir yargılayıcı tutum sergileyen yorumlar da olabilir. Bu gibi durumlarda Admin sayfanın düzeni için bu tarzdaki yorumları kaldırma hakkına sahiptir. Bu yüzden bu özellik yüksek önceliklidir diyebiliriz. Bu özellik sayesinde bloğumuzun yorum kalite seviyesini düşürmemiş oluruz ve diğer ziyaretçiler için herhangi bir sorun teşkil etmesini engellemiş oluruz.

#### Uyaran/Yanıt Dizileri

Bu özellik için tanımlanan davranışı uyaran kullanıcı eylemlerinin ve sistem yanıtlarının sırasını listeleyin. Bunlar, kullanım durumları ile ilişkili diyalog öğelerine karşılık gelir.

Yapılan yorumlar sadece admin sayfası tarafından silinebilir. Bu yüzden ilk olarak sisteme admin panelinden giriş yapılmalıdır. Daha sonrasında ise yorumlar kısmından ilgili yorumu bulup silme butonuna basılmalıdır. Sistem bu işlemden sonra datada ilgili tablodan o yorumu silmiş olur. Böylece ilgili bloğun yorumlar kısmından da silinmiş olur.

#### İşlevsel Gereklilikler

Bu özellikle ilişkili ayrıntılı işlevsel gereksinimleri maddeleyin. Bunlar, kullanıcının özellik tarafından sağlanan hizmetleri yerine getirmesi veya kullanım senaryosunu yürütmesi için mevcut olması gereken yazılım yetenekleridir. Ürünün beklenen hata koşullarına veya geçersiz girdilere nasıl yanıt vermesi gerektiğini dahil edin. Gereksinimler özlü, eksiksiz, açık, doğrulanabilir ve gerekli olmalıdır. Gerekli bilgilerin henüz mevcut olmadığını belirtmek için “TBD”yi (to-be-determined) kullanın. Her gereksinim, bir sıra numarası veya bir tür anlamlı etiketle (REQ-1, REQ-2, …) benzersiz bir şekilde tanımlanmalıdır.

Silinmek istenen yoruma admin panelinden ulaşılamaz ise veritabanından silinmesi gerekebilir.

### < Ziyaretçilerin admine mesaj atmaları >

Gerçekten “Sistem Özelliği 1” demeyin. Özellik adını birkaç kelimeyle belirtin.

#### Tanım ve Öncelik

Özelliğin kısa bir tanımını yapın ve Yüksek, Orta veya Düşük öncelikli olup olmadığını belirtin. Ayrıca fayda, ceza, maliyet ve risk gibi belirli öncelikli bileşen derecelendirmelerini de dahil edebilirsiniz (her biri en düşük 1 ile en yüksek 9 arasında göreceli bir ölçekte derecelendirilmiştir).

Ziyaretçiler herhangi bir istek, şikayet, düşünce, admin olma isteği vb. gibi konular için admine mesaj atabilmektedirler. Bu özellik orta seviye önceliklidir. Bu özellik, web sitesinde kullanıcı isteğinin ön planda olduğunu gösterir.

#### Uyaran/Yanıt Dizileri

Bu özellik için tanımlanan davranışı uyaran kullanıcı eylemlerinin ve sistem yanıtlarının sırasını listeleyin. Bunlar, kullanım durumları ile ilişkili diyalog öğelerine karşılık gelir.

Bu işlem web sitesindeki iletişim kısmında bulunan formun doldurulması ve gönder butonuna basılması ile gerçekleşir. Butona basıldıktan sonra gönderilen mesaj admin panelindeki iletişim kısmına düşmüş olur.

#### İşlevsel Gereklilikler

Bu özellikle ilişkili ayrıntılı işlevsel gereksinimleri maddeleyin. Bunlar, kullanıcının özellik tarafından sağlanan hizmetleri yerine getirmesi veya kullanım senaryosunu yürütmesi için mevcut olması gereken yazılım yetenekleridir. Ürünün beklenen hata koşullarına veya geçersiz girdilere nasıl yanıt vermesi gerektiğini dahil edin. Gereksinimler özlü, eksiksiz, açık, doğrulanabilir ve gerekli olmalıdır. Gerekli bilgilerin henüz mevcut olmadığını belirtmek için “TBD”yi (to-be-determined) kullanın. Her gereksinim, bir sıra numarası veya bir tür anlamlı etiketle (REQ-1, REQ-2, …) benzersiz bir şekilde tanımlanmalıdır.

Form panelinde doldurulması gereken alanları doldurmadan form atılırsa uyarı mesajı çıkar. O alanları doldurmadan admine mesaj atamazlar.

### < Adminin blogları yönetmesi >

Gerçekten “Sistem Özelliği 1” demeyin. Özellik adını birkaç kelimeyle belirtin.

#### Tanım ve Öncelik

Özelliğin kısa bir tanımını yapın ve Yüksek, Orta veya Düşük öncelikli olup olmadığını belirtin. Ayrıca fayda, ceza, maliyet ve risk gibi belirli öncelikli bileşen derecelendirmelerini de dahil edebilirsiniz (her biri en düşük 1 ile en yüksek 9 arasında göreceli bir ölçekte derecelendirilmiştir).

Adminler sayfaya blog ekleyebilir, bloğu düzenleyebilir ve bloğu silebilir. Bu özellik yüksek önceliklidir. Çünkü bu özellik olmadığında web sitesi amacına uygun olarak var olamaz.

#### Uyaran/Yanıt Dizileri

Bu özellik için tanımlanan davranışı uyaran kullanıcı eylemlerinin ve sistem yanıtlarının sırasını listeleyin. Bunlar, kullanım durumları ile ilişkili diyalog öğelerine karşılık gelir.

Bu işlemlerin gerçekleşmesi için öncelikle admin panelinden giriş yapılmalıdır. Daha sonrasında bloglar kısmında ekle butonuna basarak blog ekleyebilir. Listede var olan blogların yanındaki butonlardan bloğu silebilir ya da düzenleyebilir. Bu yapılan işlemler aynı şekilde database’de de gerçekleşir.

#### İşlevsel Gereklilikler

Bu özellikle ilişkili ayrıntılı işlevsel gereksinimleri maddeleyin. Bunlar, kullanıcının özellik tarafından sağlanan hizmetleri yerine getirmesi veya kullanım senaryosunu yürütmesi için mevcut olması gereken yazılım yetenekleridir. Ürünün beklenen hata koşullarına veya geçersiz girdilere nasıl yanıt vermesi gerektiğini dahil edin. Gereksinimler özlü, eksiksiz, açık, doğrulanabilir ve gerekli olmalıdır. Gerekli bilgilerin henüz mevcut olmadığını belirtmek için “TBD”yi (to-be-determined) kullanın. Her gereksinim, bir sıra numarası veya bir tür anlamlı etiketle (REQ-1, REQ-2, …) benzersiz bir şekilde tanımlanmalıdır.

Sayfada blog işlemlerini yapmak için admin panelinden girilmesi gerekir. Yani admin panelinden giriş yapmak zorunludur. Blog eklemesi yaparken oradaki alanların doldurulması gerekir yoksa blog eklenmez ya da eklenen blog hatalı gösterilebilir.

## Yazılım

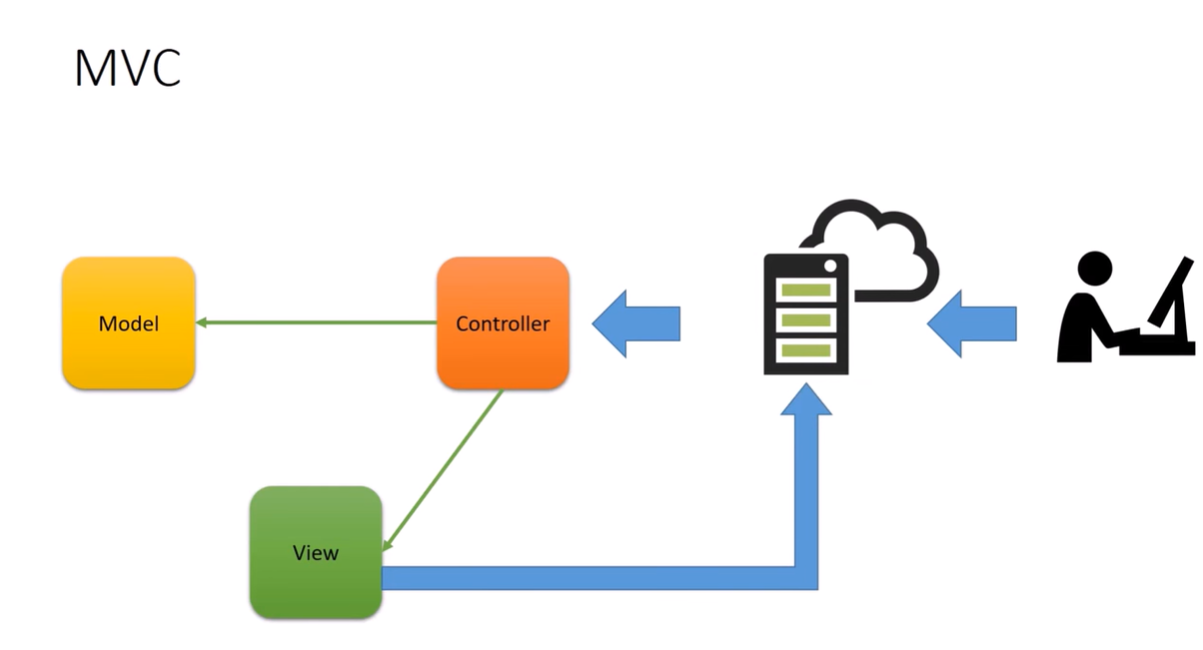
Yazılım mimarisiyle ilgili ayrıntıları gösterin. n-tier mimari kullanıyor musunuz? Projenizde nasıl tasarladınız? Yazılım mimarisi diyagramı çizebilirsiniz (Tipik bir yazılım sisteminin kullanıcıları, dış sistemleri, veri kaynakları ve hizmetleri ile nasıl etkileşime girebileceğini gösterir.). Veritabanı ile ön uç (front-end) arasındaki bağlantıyı nasıl sağladınız? Yazılım mimariniz hakkında ayrıntılı bilgi verin.

Projemizde n-tier mimari kullandık. Bunu **Asp.Net MVC** teknolojisi ile gerçekleştirdik.

Html, css, javascript gibi **front-end** kodlarını **View** katmanında oluşturduk. Websitesinin her bir sayfasını bu cshtml dosya uzantılı View’lar oluşturur.

Veri tabanı bağlantısını yani **back-end** tarafını **Model** katmanında oluşturduk. Bu katmana cs dosya uzantılı sınıflar oluşturduk. Bu sınıflar, veritabanındaki her bir tabloya karşılık gelir.

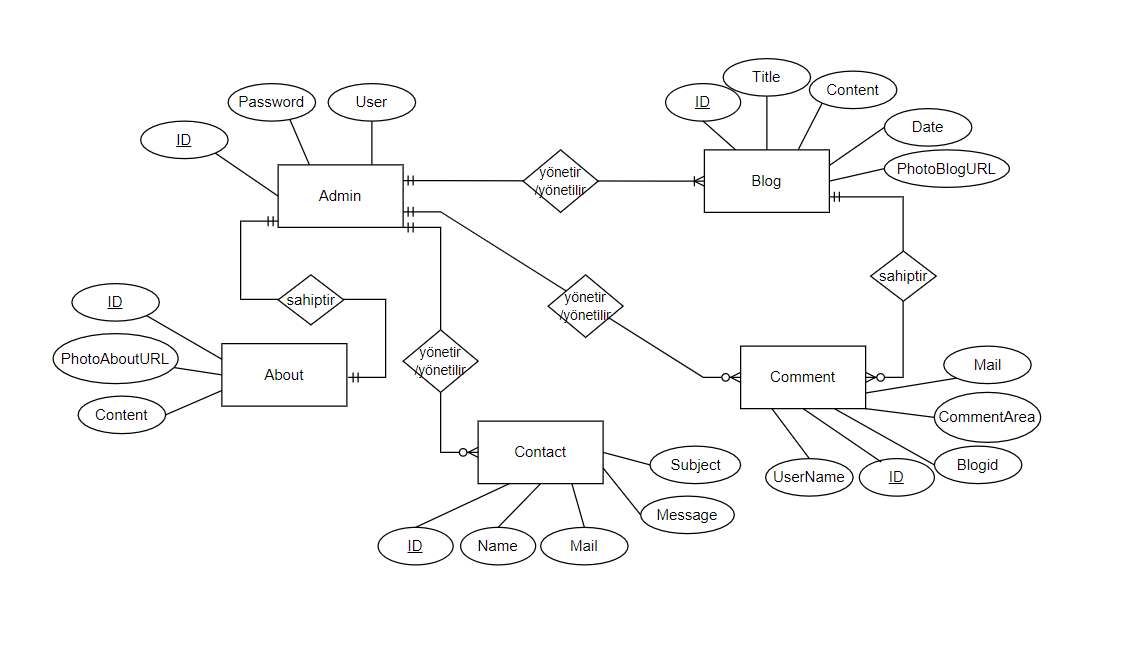
Kullanıcı isteklerinin karşılanmasını da **Controller** katmanında gerçekleştirdik. Bu katmanda veritabanı (**back-end**) ile kullanıcı arayüzü (**front-end**) arasındaki ilişkiyi sağladık.



## Sistem Tasarımı

### Varlık-İlişki Diyagramı

*Tüm kuralları (tablolar, özellikler, ilişkiler, kardinaliteler, vb.) uygulayarak Varlık İlişki Diyagramını çizin.*

**

### İlişkisel Cebir İfadeleri

*İlişkisel cebir ifadeleri ve onların SQL cümleleri*

1) Bloglara yorum yapanların isimmleri ve mail adresleri.

ΠName,Mail*(Comments)*

* 1. SQL cümlesi:

SELECT Comments.UserName, Mail FROM Comments

2) 1 nolu id olan bloğa yorum yapanların isimmleri ve yorumları.

ΠName,Mail *(*σID=”1”*(Comments))*

2.1) SQL cümlesi:

SELECT Comments.UserName, Mail FROM Comments WHERE ID=1

3) Meksika başlıklı bloğa yorum yapanların isimmleri.

ΠName *(Comments* **⋈** σTitle=”Meksika”*(Blogs))*

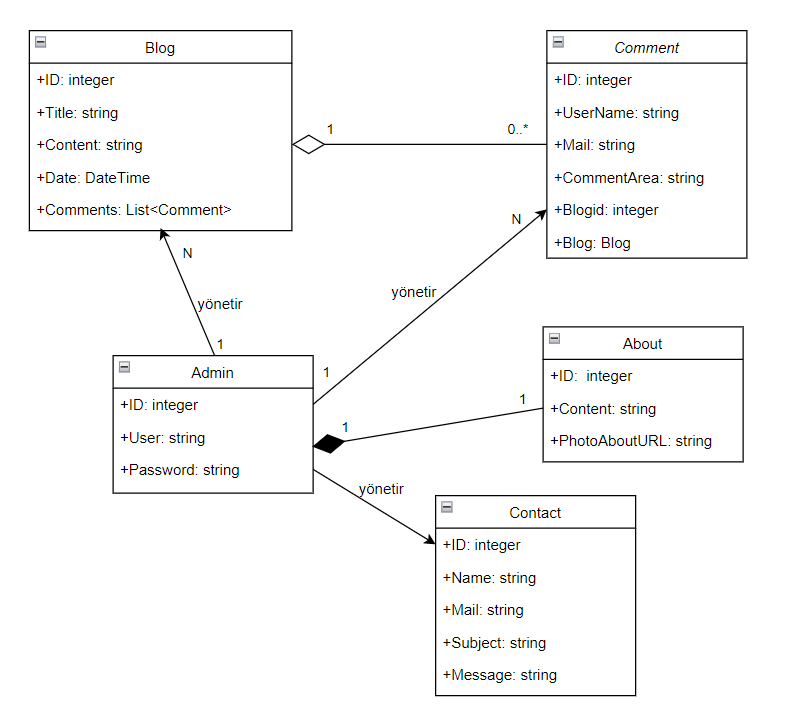
3.1) SQL cümlesi:

select Comments.UserName,Comments.Blogid

from Comments

where ID in (select ID from Blogs where Title='Meksika')

### Sınıf Diyagramı



### *CRUD* Matrisi

Aşağıdaki tablodaki gibi tüm ilişkiler için verilerin nasıl korunacağını ve verilere nasıl erişileceğini gösteren CRUD Matrisi (Create, Read, Update, Delete) oluşturun. Tüm CRUD işlemlerini tüm tablolara uyguladınız mı?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tablo-Operasyon** | **Blogs** | **Admins** | **Abouts** | **Comments** | **Contact** |
| **Blog sayfası** | R | CDU |  |  |  |
| **Yorum yapmak** | C |  |  | C |  |
| **Yorum silme** |  | D |  | D |  |
| **İletişim sayfası** |  | CDU | R |  | R |
| **Giriş yapma/çıkma** |  | R |  |  |  |

### View, Trigger ve Stored Procedure Kullanımı

*Kullandığınız view, trigger ve stored procedure kullanımlarını açıklayınız.*

Kullandığımız **view** yorumların tarihleriyle beraber gösterilmesi ile alakalıdır.

İki tane **trigger** kullandık. Bu iki trigger aynı amaca yönelik olarak, sitedeki toplam yorumların sayısını tutar.

1.trigger; web sayfasındaki herhangi bir bloğa her yorum eklemede counter tablosundaki commentscounter’ı 1 artırır.

2.trigger; web sayfasındaki herhangi bir blogtan her yorum silmede counter tablosundaki commentscounter’ı 1 azaltır. Bu iki trigger ile yorumların eklenmesi ya da silinmesi dinamik olarak toplam yorum sayısına ulaşabiliriz.

Dört tane **stored procedure** kullandık.

1.procedure; blogs tablosundaki blogların listelenmesidir.

2.procedure; blogs tablosuna yeni blog eklenirken bloğa ait özelliklerin alınarak bloğun oluşturulmasıdır.

3.procedure; blogs tablosundaki bir bloğun özelliklerinin update ile güncellenmesidir.

4.procedure; blogs tablosundaki bir bloğun id’sine bağlı olarak veritabanından silinmesidir.

# Ek A: SQL İfadeleri

Tüm SQL ifadelerine yer verin (komut dosyaları oluşturun ve ekleyin, veritabanındaki temel işlemler, view’ler, karmaşık sorgular, stored procedure’ler, trigger’lar)

USE [master]

GO

/\*\*\*\*\*\* Object: Database [GeziLogDb] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

CREATE DATABASE [GeziLogDb]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'GeziLogDb', FILENAME = N'D:\Sql Server 2019\MSSQL15.MSSQLSERVER\MSSQL\DATA\GeziLogDb.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'GeziLogDb\_log', FILENAME = N'D:\Sql Server 2019\MSSQL15.MSSQLSERVER\MSSQL\DATA\GeziLogDb\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

ALTER DATABASE [GeziLogDb] SET COMPATIBILITY\_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [GeziLogDb].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [GeziLogDb] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [GeziLogDb] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [GeziLogDb] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [GeziLogDb] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [GeziLogDb] SET ARITHABORT OFF

GO

ALTER DATABASE [GeziLogDb] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [GeziLogDb] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [GeziLogDb] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [GeziLogDb] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [GeziLogDb] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [GeziLogDb] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [GeziLogDb] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [GeziLogDb] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [GeziLogDb] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [GeziLogDb] SET ENABLE\_BROKER

GO

ALTER DATABASE [GeziLogDb] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [GeziLogDb] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [GeziLogDb] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [GeziLogDb] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [GeziLogDb] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [GeziLogDb] SET READ\_COMMITTED\_SNAPSHOT ON

GO

ALTER DATABASE [GeziLogDb] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [GeziLogDb] SET RECOVERY FULL

GO

ALTER DATABASE [GeziLogDb] SET MULTI\_USER

GO

ALTER DATABASE [GeziLogDb] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [GeziLogDb] SET DB\_CHAINING OFF

GO

ALTER DATABASE [GeziLogDb] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [GeziLogDb] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [GeziLogDb] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [GeziLogDb] SET ACCELERATED\_DATABASE\_RECOVERY = OFF

GO

EXEC sys.sp\_db\_vardecimal\_storage\_format N'GeziLogDb', N'ON'

GO

ALTER DATABASE [GeziLogDb] SET QUERY\_STORE = OFF

GO

USE [GeziLogDb]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Blogs] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Blogs](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Title] [nvarchar](max) NULL,

[Content] [nvarchar](max) NULL,

[Date] [datetime] NOT NULL,

[PhotoBlogURL] [nvarchar](max) NULL,

CONSTRAINT [PK\_dbo.Blogs] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Comments] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Comments](

[ID] [int] IDENTITY(1,1) NOT NULL,

[UserName] [nvarchar](max) NULL,

[Mail] [nvarchar](max) NULL,

[CommentArea] [nvarchar](max) NULL,

[Blogid] [int] NOT NULL,

CONSTRAINT [PK\_dbo.Comments] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: View [dbo].[COMMENTDATES] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[COMMENTDATES]

AS

SELECT dbo.Blogs.Date, dbo.Comments.CommentArea

FROM dbo.Blogs INNER JOIN

dbo.Comments ON dbo.Comments.Blogid = dbo.Blogs.ID

GO

/\*\*\*\*\*\* Object: Table [dbo].[\_\_MigrationHistory] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[\_\_MigrationHistory](

[MigrationId] [nvarchar](150) NOT NULL,

[ContextKey] [nvarchar](300) NOT NULL,

[Model] [varbinary](max) NOT NULL,

[ProductVersion] [nvarchar](32) NOT NULL,

CONSTRAINT [PK\_dbo.\_\_MigrationHistory] PRIMARY KEY CLUSTERED

(

[MigrationId] ASC,

[ContextKey] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Abouts] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Abouts](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Content] [nvarchar](max) NULL,

[PhotoAboutURL] [nvarchar](max) NULL,

CONSTRAINT [PK\_dbo.Abouts] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Admins] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Admins](

[ID] [int] IDENTITY(1,1) NOT NULL,

[User] [nvarchar](max) NULL,

[Password] [nvarchar](max) NULL,

CONSTRAINT [PK\_dbo.Admins] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Contacts] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Contacts](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](max) NULL,

[Mail] [nvarchar](max) NULL,

[Subject] [nvarchar](max) NULL,

[Message] [nvarchar](max) NULL,

CONSTRAINT [PK\_dbo.Contacts] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Counters] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Counters](

[CounterID] [int] IDENTITY(1,1) NOT NULL,

[CommentCounter] [int] NOT NULL,

CONSTRAINT [PK\_dbo.Counters] PRIMARY KEY CLUSTERED

(

[CounterID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Index [IX\_Blogid] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [IX\_Blogid] ON [dbo].[Comments]

(

[Blogid] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Comments] WITH CHECK ADD CONSTRAINT [FK\_dbo.Comments\_dbo.Blogs\_Blogid] FOREIGN KEY([Blogid])

REFERENCES [dbo].[Blogs] ([ID])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Comments] CHECK CONSTRAINT [FK\_dbo.Comments\_dbo.Blogs\_Blogid]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[BLOGEKLE] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[BLOGEKLE]

@Title nvarchar(MAX),

@Content nvarchar(MAX),

@Date datetime,

@Photo nvarchar(MAX)

AS

INSERT INTO Blogs(Title, Content, Date, PhotoBlogURL)

VALUES (@Title, @Content, @Date, @Photo)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[BLOGGUNCELLE] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[BLOGGUNCELLE]

@Title nvarchar(MAX),

@Content nvarchar(MAX),

@Date datetime,

@Photo nvarchar(MAX),

@ID int

AS

UPDATE Blogs SET

Title = @Title,

Content = @Content,

Date = @Date,

PhotoBlogURL = @Photo

WHERE ID = @ID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[BLOGLISTESI] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[BLOGLISTESI]

AS

SELECT \* FROM Blogs

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[BLOGSIL] Script Date: 5.01.2022 03:17:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[BLOGSIL]

@ID int

AS

DELETE FROM Blogs WHERE ID = @ID

USE [GeziLogDb]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[COMMENTCOUNTER] Script Date: 3.01.2022 15:08:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[COMMENTCOUNTER]

ON [dbo].[Comments]

AFTER INSERT

AS

UPDATE Counters

SET CommentCounter = CommentCounter + 1

WHERE CounterID= 1

USE [GeziLogDb]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[COMMENTCOUNTER2] Script Date: 3.01.2022 15:08:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[COMMENTCOUNTER2]

ON [dbo].[Comments]

AFTER DELETE

AS

UPDATE Counters

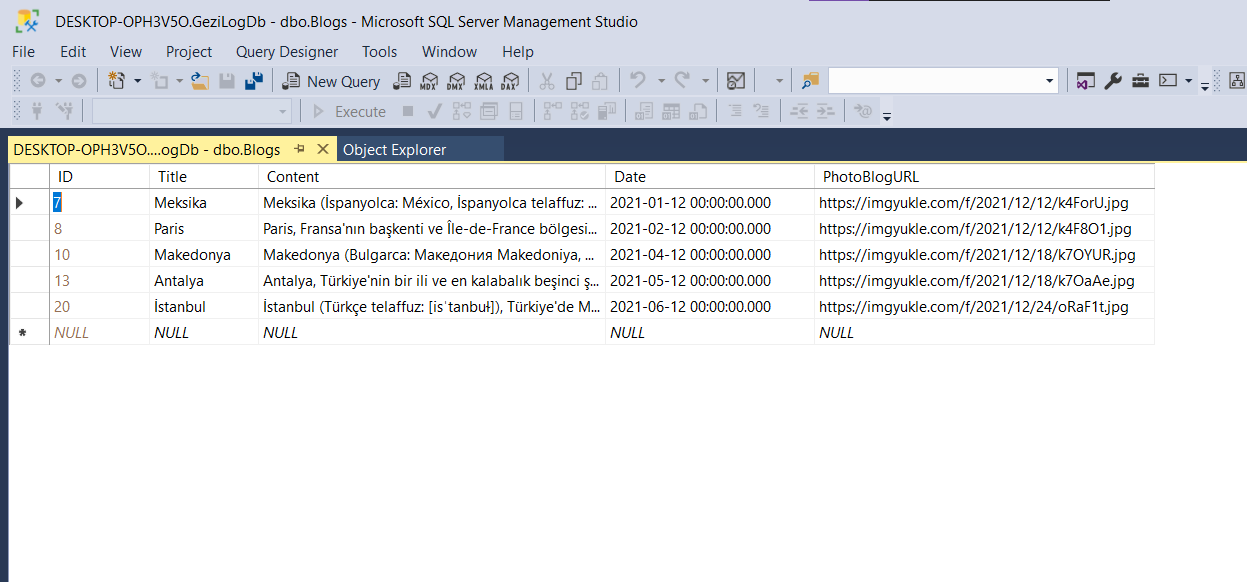
SET CommentCounter = CommentCounter - 1

WHERE CounterID= 1

# Ek B: Ekran Görüntüleri

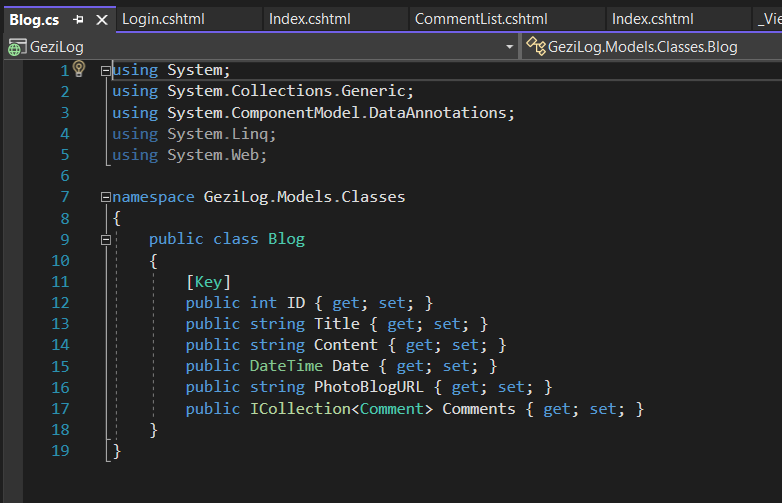
*Ana işlemler için kısa açıklamalı arayüzlerinizin ekran görüntüleri*

SQL Management Studio: GeziLogDB veritabanını ve bu veritabanında Blogs adlı tabloyu oluşturduk. Blogs tablosuna 5 tane veri ekledik.

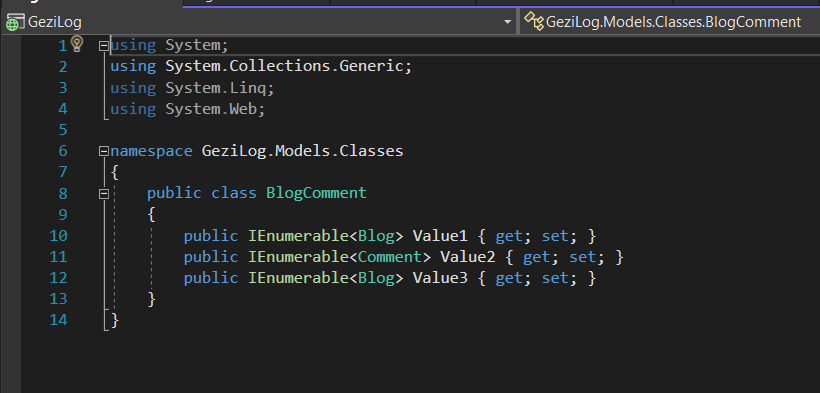
**

Visual Studio:

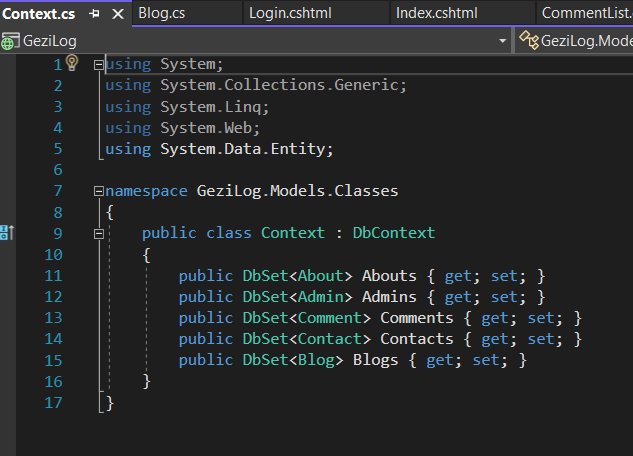
Adım1: Blog sınıfı oluşturduk ve özelliklerini verdik.

**

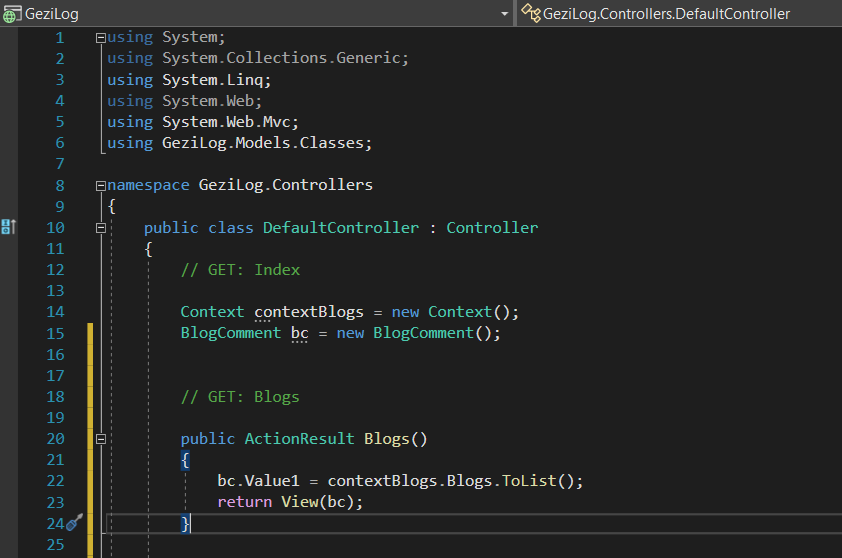
Adım2: BlogComment sınıfı oluşturup, Blog’ları Value1 adındaki listede tuttuk.

**

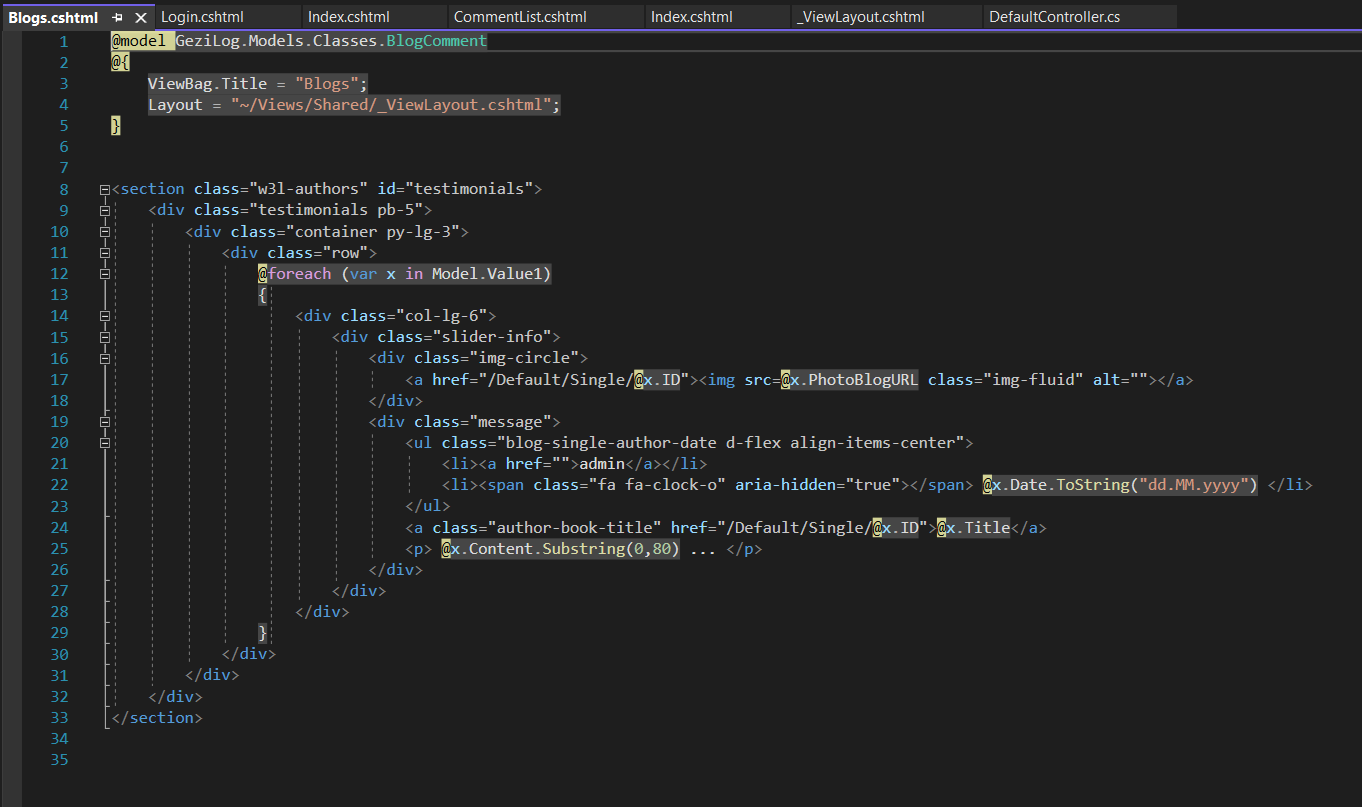
Adım3: Context sınıfı oluşturup bu sınıfı DbContext sınıfından miras aldık. Böylece veri tabanındaki tablolarla bağlantı kurduk.

**

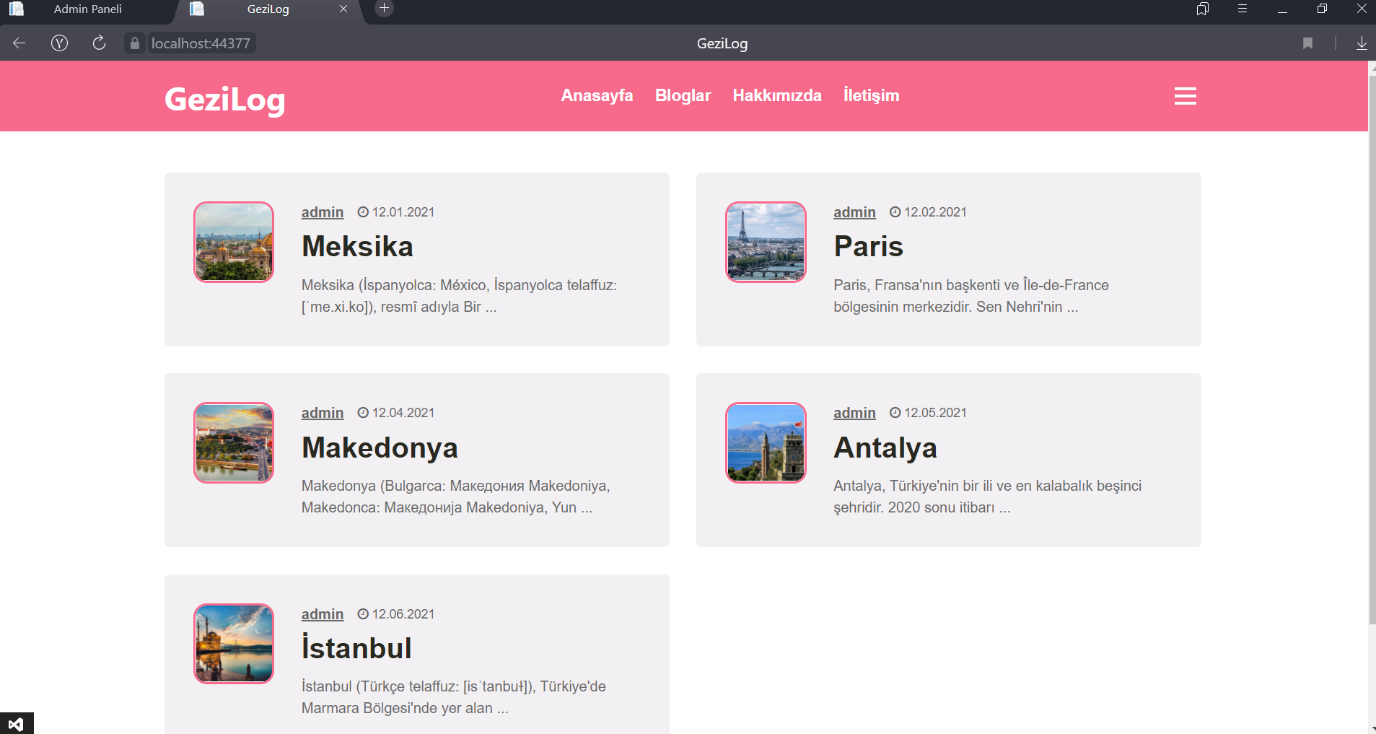
Adım4: Controller tarafında; veritabanında bulunan Blogs tablosundaki verilerin değerlerini, Value1 listesindeki değerlere atadık.

**

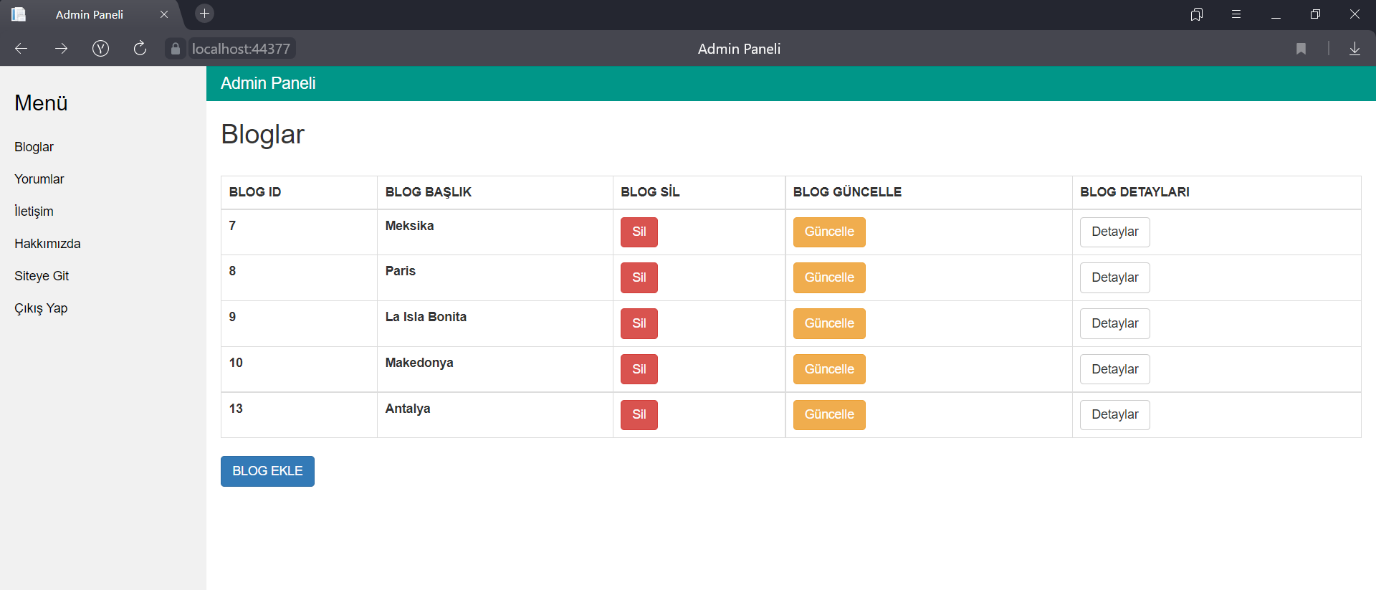
Adım5: Kullanıcı tarafında; html kodlarıyla oluşturduğumuz site arayüzüne, Value1 listesine atadığımız değerleri foreach döngüsü yardımıyla çektik.

**

Websitesi: Bloglar listelendi.

**

Admin Paneli: Admin ise admin panelinden bu bloglara erişebilir.

**

# Ek C: To Be Determined List (TBDL)