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

**“Small” Blog Sosyal Medya Platformu**

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

2021-2022 GÜZ

BİL 203 & BİL 209 Dönem Projesi

Sürüm 3.0

*Hazırlayan*

200601046 Sedat KORKMAZ

200601031 Alperen ÇORAK

İçindekiler

İçindekiler 2

Revizyon Geçmişi 2

1. Giriş 3

1.1 Amaç 4

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 raporu şablonunun oluşturulması ve açıklanması | 1.0 |
| Faz 1 | 30.11.2021 | Proje gereksinimleri ve analizi yapıldı. | 1.1 |
| Faz 2 | 18.12.2021 | Gereksinim ve analizler güncellendi. CRUD operasyonları ve arayüzleri paylaşıldı. UML diagramları güncellendi. Yazılım mimarisi hakkında bilgi verildi. | 2.0 |
| Final | 05.01.2022 | Ekran görüntüleri eklendi. İlişkisel cebir işlemleri ve bunların SQL/NoSQL cümleleri eklendi. View, trigger ve stored procedurelar verildi. MongoDB ifadeleri verildi. Sistem özelliklerinden bahsedildi. Arayüzlerden bahsedildi. Rapor son haline getirildi | 3.0 |

## Giriş

Bu bölümde önerilen sistem için tanımlayıcı bilgilere yer verilir.

Small bir blog sosyal medya platformudur. Bu sistem kullanıcıların kendi blog yazılarını paylaşabildiği ve diğer kullanıcıların içerikleri ile etkileşime geçebildiği bir platformdur.

### 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.

Small blog sosyal medya platformu, kullanıcıların kim olduğuna bakmaksızın bu kullanıcıların bilgi birikimleri ile eşsiz fikirlerini bağımsızca yazıp dünya üzerindeki herkesle paylaşmasını amaçlamaktadır. Bu paylaşılan fikirler ile insanların birbirini anlamasını, keşfetmesini ve düşüncülerini zenginleştirmesini istiyoruz.

Bu rapor ise aşağıda belirtilen hedef okuyuculara bahsedilen sistemin gereksinimlerini ve özelliklerini raporlamak için yazılmıştır.

* Yatırımcılar
* Proje Yöneticileri
* Veri Tabanı Geliştiricileri
* Test Mühendisleri
* Yazılım Mühendisleri
* Sistem Yöneticileri
* Pazarlamacılar

### 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.

Bu rapor başlı başına bir sistemi açıklamaktadır. Sözü edilen bu sistem bir blog sosyal medya platformu olup birden fazla modülü içinde barındırır.

Bu blog sisteminde kullanıcıların birbirleriyle bilgi paylaşımını kolay ve erişilebilir şekilde yapmalarını sağlamak istiyoruz.

Blog kısmında kayıtlı kullanıcılar istediği bir konuda blog yazabilir, düzenleyebilir ve bunu public bir şekilde paylaşabilir. Bu yazılan blog hem site ziyaretçisi tarafından hem de diğer kayıtlı kullanıcılar tarafından görüntülenebilir fakat bloğu kaydetme, beğenme özellikleri sadece kayıtlı kullanıcılara özeldir.

Anasayfa kısmında kayıtlı kullanıcılar ve site ziyaretçileri son yazılan ve popüler blogları görüntüleyebilirler.

Arama kısmında kayıtlı kullanıcılar ve site ziyaretçileri blogları arayıp listeleyebilirler.

### 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.

Kurallar : Bu rapor içerisinde gereksinimler belirli bir önem sırasına göre sıralanır. Vurgulanmak istenen tanımlar kalın harflerle yazılır.

Kısaltmalar :

|  |  |
| --- | --- |
| KVKK | Kullanıcı veri koruma kanunu |
| DAL | Data Access Layer |
| API | Application programming interface |
| HTTP | Hyper Text Transfer Protocol |
| CRUD | Create Read Update Delete |
| MVC | Model View Controller |

### 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.

### 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.

Bu bölümden sonra rapor 2. Bölümde uygulama gereksinimlerine ve işlevlerine genel bir bakış atılır ve bilgi verilir. 2. Bölüm bu sebeple tüm okuyuculara hitap etmektedir.

Geliştiriciler 3. Bölümden okumaya başlayabilir. 3. Bölüm uygulamanın yazılımı hakkında bilgiler verir.

4. bölüm tüm okuyuculara hitap etmekte olup geliştiriciler içinde ifadeler içermektedir. Sistemin genel özelliklerinden bahsedilir.

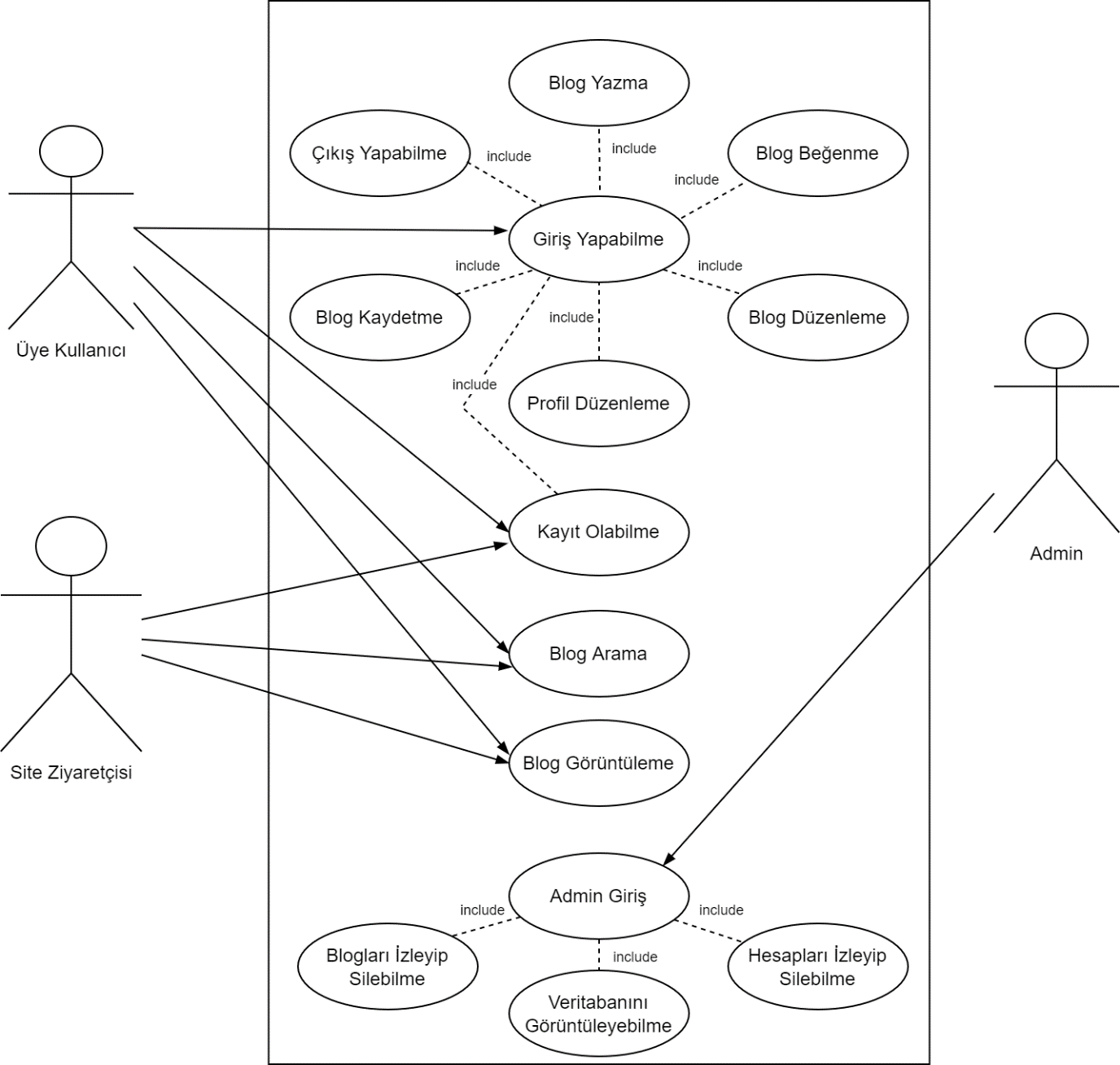
## Genel Tanım

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.

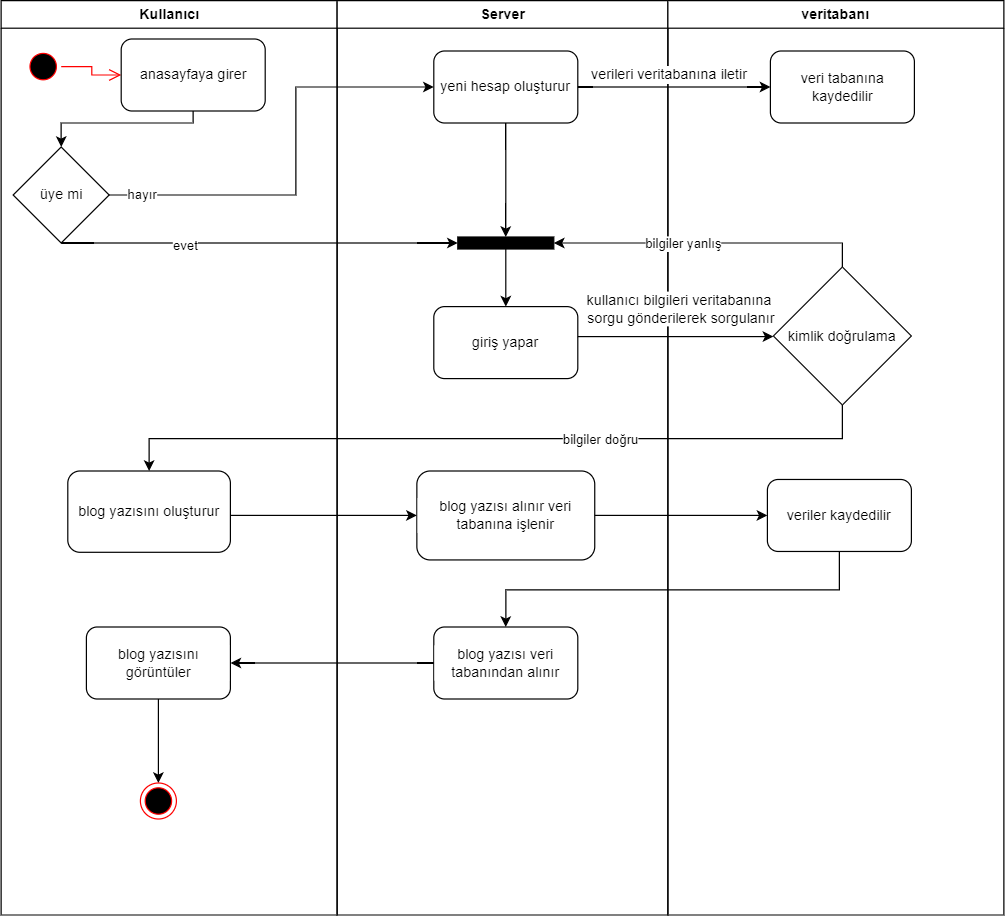
Sistem içeriği genel olarak giriş, kayıt olma, anasayfa, kullanıcı profili, blog oluşturma sayfalarından oluşur. Sistem üyeler için giriş yapabilme, blog oluşturabilme, blog düzenleyebilme, blog kaydedebilme, blog beğenebilme, profil düzenleyebilme, blog arayabilme, blog görüntüleyebilme ve çıkış yapabilme işlemlerini sağlar. Site ziyaretçileri için kayıt olabilme, blog arayabilme, blog görüntüleme işlemlerini sağlar. Adminler için admin girişi yapabilme, blogları izleyip silebilme, veritabanı görüntüleyebilme, hesapları izleyip silebilme işlemlerini sağlar.

Sistem ek olarak anasayfa üzerinde popüler ve en son yüklenen blogları listeleyebilir. 

**Örnek kullanım senaryoları;**

Bütün kullanıcılar anasayfayı görüntüleyebilir ve burdaki blogları okuyabilir. Aynı zamanda arama sekmesi üzerinden aradığı konuyla alakalı server’a istekte bulunabilir.

Üye kullanıcılar kendi bloglarını yazıp server’a istekte bulunup bu blogları kendi profilleri üzerinden görüntüleyebilirler.

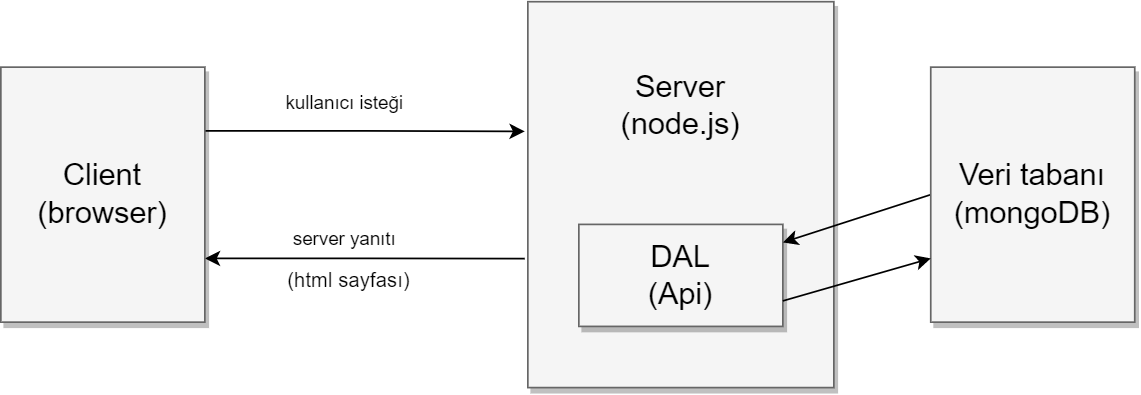
Adminler uygunsuz gördüğü bir profilin erişimini engelleyebilir. Aynı şekilde uygunsuz gördüğü bir blogu silebilir.

Yukarıda en temel özelliklerden biri olan bir kullanıcının blog paylaşma süreci aktivite diyagramı kullanılarak açıklanmaya çalışılmıştır.

### 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.

Small blog sosyal medya platformu başlı başına bağımsız bir web uygulamasıdır.

Yukarıdaki blok diyagramında bu web uygulamasının genel hatları ile nasıl çalışacağını göstermektedir. Bu web uygulaması 3 temel parçadan oluşmaktadır. Bu parçalar Client (web browser), server ve veri tabanıdır. Genel olarak çalışma mantığı ise;

* Kullanıcı HTTP olarak servera istekte bulunur.
* Bu istek server tarafından değerlendirilir.
* Server api aracılığı ile veri tabanına bağlanır ve gerekli sorguları gerçekleştirilir.
* Bu sorgular sonucunda kullanıcı isteğine göre bir web sayfası oluşturulur ve browsera iletilir.

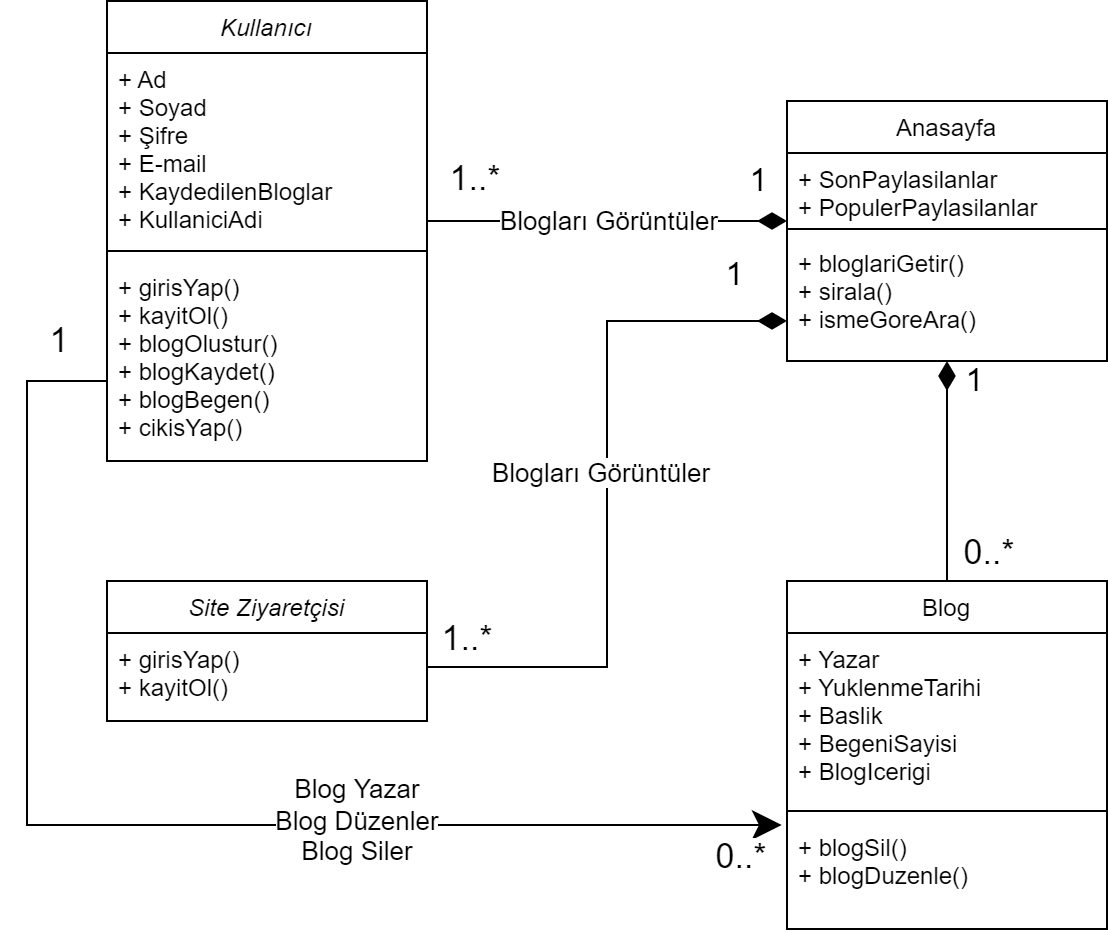
### Ürün İşlevleri

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.

* Kullanıcılar sisteme kaydolabilir.
* Kayıtlı kullanıcılar sisteme giriş yapabilir.
* Üye olmayan site ziyaretçileri ve kayıtlı kullanıcılar blog görüntüleyebilir ve arama yapabilir.
* Kayıtlı kullanıcılar blog yazabilir ve bu blogları düzenleyebilir, silebilir ve paylaşabilir.
* Kayıtlı kullanıcılar blogları beğenebilir ve kaydedebilir. Bu kaydettiği bloglara profilleri üzerinden erişebilir.
* Kayıtlı kullanıcılar profillerini istedikleri gibi düzenleyebilirler.

(Admin işlemleri şuan sadece API üzerinden gerçekleştirilebilecek olup güncellenecektir.)

* Adminler blogları ve hesapları izleyebilir uygun görmediklerini silebilirler.
* Adminler veritabanını görüntüleyebilirler.



Ürünü işlevleri açıklanması ve görselleştirilmesi açısından yukarıdaki sınıf diyagramı çizilmiştir. Bu sınıf diyagramında görüldüğü üzere üründen beklenen en temel işlevler çizilerek gösterilmiştir.

### 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.

Bu ürün için belirlenen en temel üç kullanıcı vardır. Bunlar; Yazarlar, okuyucular ve sistem yöneticileridir (adminler). Yazarlar sınıfı bu ürünü büyük çoğunlukla blog yazmak için kullanan kullanıcılar olarak nitelendirilmektedir. Okuyucular çoğunlukla yazılan blogları okuyup kaydetmek için bu ürünün kullanan kullanıcılar olarak nitelendirilmektir. Sistem yöneticileri bu sistemin yöneticileri olup sistem üzerinden veri tabanına erişim sağlayıp bu sistemin sorunsuz ve kurallara uygun işlemesi için kullanan kullanıcılar olarak nitelendirilmektedir.

### 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.

Ürün bir web uygulaması olup tarayıcılar üzerinden çalışmaktadır. Ürün geliştirilirken Chrome, Firefox, Opera ve Edge tarayıcılarında test edilmiştir. Bu tarayıcılar üzerinde sorunsuz çalışmakta olup farklı tarayıcılarda da çalışabilir. Web sitelerine erişimi olan tüm cihazlarda çalışabilir.

### 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).

Bölüm 2.1 de uygulama tasarımı hakkında bilgi verilmiştir. Uygulama geliştirilirken kullanılar teknolojiler, uygulamalar ve donanım aşağıda belirtilmiştir.

**Donanım**

-İşletim sistemi: windows 10, windows 11

-Ram: 16gb

İşlemci: i5 9300h

**Kısıtlamalar**

* **Kullanıcı Kısıtlamaları**

Bu uygulamada kullanıcı verileri toplanacağı için KVKK gereklilikleri bulunmaktadır. Kullanıcılar bu konuda bilgilendirilecektir. Kullanıcı verilerin güvenliği için bcrypt şifre hashleme algoritması kullanılmıştır. Kullanıcı giriş işlemleri için token oluşturulup çerezlere kaydedilmektedir. Bu sebeple kullanıcılardan çerez kullanımına izin verilmesi istenmektedir.

* **Geliştirici Kısıtlamaları**

Uygulamanın çalıştırılması hakkındaki gereklilikler ReadMe dosyasında belirtilmiştir.

***API için gerekli olan yazılım ve uygulamalar;***

API’ın geliştirme süreci için Nodejs kullanılmıştır. Bu Nodejs uygulamasının çalıştırılması için proje içinde bulunan *package.json* dosyasında belirtilen gerekli paketlerin yüklenilmesi gerekmektedir. Bu paketler;

- Express

- Bcrypt

- Dotenv

- ExpressAsyncHandler

- JsonWebToken

- Mongoose

- Multer

- Slugify

- Cors

Bu paketler yüklendikten sonra API’ın database ile bağlantısının yapılması gerekmektedir. Bu sebeple MongoDB Connection String’i girilmelidir.

***Client için gerekli olan yazılım ve uygulamalar;***

Client Side’ın geliştirme süreci için Nodejs ve Ejs kullanılmıştır. Bu uygulamanın çalıştırılması için proje içinde bulunan *package.json* dosyasında belirtilen gerekli paketlerin yüklenmesi gerekmektedir. Bu paketler;

* Axios
* Dotenv
* Ejs
* Express
* ExpressAsyncHandler
* Lodash
* Multer
* Form-Data

Bu paketler yüklendikten sonra API ile Client Side’ın bağlantısının yapılması gerekmektedir. Bu sebeple API Url’inin *Config* dosyasına girilmesi gerekmektedir.

### 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.

Bu geliştirilen yazılım ile birlikte bir Github adresi geliştiricilere verilecektir. Bu Github adresinde bulunan *Readme* dosyasında projenin nasıl çalıştırılacağına dair bilgiler ve API dokümanı bulunacaktır. Bu belgeler geliştiricilere yöneliktir.

### 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.

Uygulama geliştirilirken Front-end kısmında bootstrap css kütüphanesi, backend kısmında ise nodejs ve node paketleri kullanılmıştır. Proje bu kütüphane ve paketlere ek olarak kullanılan işletim sistemi ve donanıma bağlılığı söz konusudur.

Varsayılan olarak kullanıcıların kendi epostalarıyla kaydolduğu varsayılmıştır. Ekstra bir e-posta onaylaması yapılmamaktadır.

### Riskler

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

* Sistem üzerinde bir güvenlik açığı olması.
* Kullanıcı tokenlerinin ele geçirilmesi.
* Serverların çok fazla isteğe karşılık verememesi yetersiz kalması
* Kaynak yetersizliği
* Kötü tasarım ve geliştiriciler arası iletişim sorununa bağlı olarak oluşan buglar
* UI/UX hatalar
* Oluşabilecek veri kayıpları
* Takım çalışmasının doğru sağlanamaması sonucunda projede oluşabilecek gecikmeler
* Sisteme uygun olmayan developement life cycle kullanılması

## 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.

Kullanıcı arayüzleri (kaydolma ve giriş yapma sayfaları hariç) temel olarak 3 ana bileşenden oluşmaktadır. Bunlar header, body ve footer olarak isimlendirilmiştir.

Bu üç bileşen her sayfada bulunmaktadır. Bu bileşenlerden header aşağıdaki gibi gözükmektedir.



Header bileşeni aynı zamanda navigation bar olarak adlandıralabilir. Bu arayüz bileşeni her sayfada gözükür olup kullanıcının temel işlevlerini gerçekleştirmesini sağlar.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduYandaki fotoğrafta görüldüğü gibi bu navigation bar bileşeni kayıtlı kullanıcılar için aynı zamanda açılabilir bir menü içermektedir.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduEğer kullanıcı kayıtlı değil ise bu menü yerini yanda görülen kaydolma ve giriş yapma butonlarına bırakacaktır.

Her sayfada bir body bileşeni bulunmaktadır bu bileşen genelde sayfa üzerinde gerçekleştirilen genel işlemleri görüntülenmesi veya genel içeriklerin listelenmesi işini gerçekleştirir.

Header gibi footer da her sayfada bulunmaktadır ve değişmemektedir. Footer uygulamanın lisansı hakkında bilgi içerir.

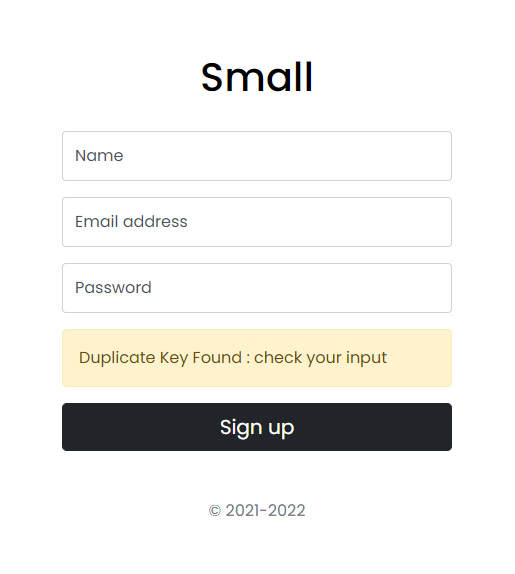
Bu uygulamada kullanılan etkileşim gerektiren temel bileşenler ise belli standartlar göz önünde bulundurularak yapılmıştır.

Yandaki ekran görüntüsü kayıtlı bir kullanıcının kendine ait bir article’ın sayfasından alınmıştır. Bu görüntüden ve yukarda gösterilen giriş yapma butonlarından da anlaşılabileceği gibi sayfa üzerindeki bütün butonlar 2 tipten oluşmaktadır. Bu butonlar siyah ve de siyah çizgilidir.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduKullanıcı arayüzleri aynı zamanda çok fazla form girdisi içermektedir. Bu formlarda belli standartlar içinde yapılmıştır.

Yukardaki görüntüde bu form girişlerinin bir örneği verilmiştir. Bu örnekte görüldüğü gibi aktif form girişlerinin kenarları siyah ve gölgelidir. Bu standart bütün form bulunan sayfalarda uygulanmıştır.

Bu form sayfalarının birçoğunda validasyon işlemi gerçekleştirilir bu işlemler sonucunda kullanıcının girdiği bilgiye göre kullanıcı uyarılır, hata mesajı gösterilir. Bu uyarı mesajı tüm uygulama boyunca aynı görünüme sahiptir.

Yanda verilen kaydolma sayfasından görüldüğü gibi bu uyarı mesajları formun sonunda sarı bir kutucuğun içinde belirtilir.

### Donanım Arayüzleri

Yazılım ürünü ile sistemin donanım bileşenleri arasındaki her arabirimin mantıksal ve fiziksel özelliklerini tanımlayın. Bu, desteklenen cihaz türlerini, verinin yapısını ve yazılım ile donanım arasındaki kontrol etkileşimlerini ve kullanılacak iletişim protokollerini içerebilir.

### 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.

Bu sistemde elektronik formlar aracılığı ile API’a istekte bulunulur ve bu istekler veritabanına işlenir.

Önyüzde gönderilen kullanıcı isteklerinde HTTP ve FTP iletişim standartlarından yararlanılmıştır. Kullanıcıdan alınan bu form verileri parse edilip body içinde tutulur. Sonrasında bu form bilgileri kullanılarak API’a istekte bulunulur.

Temel bir API isteği şu şekildedir;

Method: “GET, POST, DELETE, PUT” (Bu methodlardan sadece birini içerebilir)

Headers: Burada authentication başlığı altında token bilgisi tutulur

Body: Burada yapılacak işleme göre JSON dosyası veya Form-data objesi yollanır.

Bu istek yapısı kullanılarak API ile iletişim sağlanır. API ise geri dönüşü JSON dosyası olarak gerçekleştirir. Bu JSON dosyalarında bulunan genel objeler şu şekildedir;

{

success: true veya false

message: işlem hakkında bir bilgilendirme mesajı

data: eğer işlem sonucunda bir data dönmesi gerekiyorsa obje olarak döndürülür.

}

## 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.

### Kayıt Olma ve Giriş Yapma

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 özellik sistemde kullanıcı hesabı oluşturmaya ve giriş yapmaya yarar. Bu özellik sistemi kullanmak için ihtiyaç duyulduğundan Yüksek öncelikli bir özelliktir.

Fayda 🡪  9/9

Maliyet 🡪  7/9

Risk 🡪  7/9

#### 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.

Kullanıcı kaydolurken adını, e-mailini ve şifresini girer. Eğer emaili eşsiz değilse veya şifresi 6 karakterin altındaysa sistem uyarıda bulunur.

Giriş yaparken kullanıcı e-mailini ve şifresini girer. Girilen bilgiler kontrol edilir ve validasyon işlemi gerçekleştirilir eğer bilgiler doğru değilse sistem giriş yapmaya izin vermez.

#### 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.

* Kullanıcı kayıt olurken sisteme eşsiz bir email adresiyle kayıt olmalıdır. Sistem de bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcı kayıt olurken 6 karakterden az bir şekilde şifre belirleyemez. Sistem de bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Sistem kullanıcı şifrelerini bcrypt şifreleme algoritmasıyla şifreleyip veritabanında saklamalıdır.
* Sistem her kaydolan ve giriş yapan kullanıcı için token oluşturmalıdır.
* Sistem kullanıcı giriş yaparken girilen bilgilerin validasyonunu sağlayıp buna göre kullanıcıyı uyarmalıdır.

### Article Oluşturma

**4.2.1 Tanım ve Öncelik**

Bu özellik sistemde belli bir konu başlığı, açıklaması, içeriği ve resmi olan bir Article oluşturmaya yarar. Bu özellik sistemin bir blog sitesi olduğundan yüksek öncelikli bir özelliktir.

Fayda 🡪  9/9

Maliyet 🡪  8/9

Risk 🡪  7/9

**4.2.2 Uyaran/Yanıt Dizileri**

Kullanıcı article oluştururken başlık, açıklama, içerik yazılarını yazar ve resim ekler. Üç yazı ve resim gereklidir. Boş bırakılamaz. Başlıkların eşsiz olması gerekmektedir. Eğer başlık eşsiz değilse sistem uyarıda bulunur. Başlık ve açıklama kısımlarının yazıları 10 karakterden az olursa sistem uyarıda bulunur. İçeriğin yazısı 20 karakterden az olursa sistem uyarıda bulunur. Sistem sadece resim formatlarını kabul eder. Farklı bir dosya türü için sistem hata verir.

**4.2.3 İşlevsel Gereklilikler**

* Kullanıcının girdiği başlık eşsiz ve 10 karakterden fazla olmalıdır. Sistem bu işlemlerin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının girdiği açıklama 10 karakterden fazla olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının girdiği içerik 20 karakterden fazla olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Başlık, açıklama ve içerik yazıları gereklidir boş bırakılamaz. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Sistem üzerinde article oluşturabilmek için giriş yapılmış olması gerekmektedir. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının yüklediği resmin dosya formatı resim formatları olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.

### Article Düzenleme

**4.3.1 Tanım ve Öncelik**

Bu özellik sistemde yazılan article’ın düzenlenmesini ve silinmesini sağlar. Sistem bir blog sitesi olduğundan orta öncelikli bir özelliktir.

Fayda 🡪  9/9

Maliyet 🡪  8/9

Risk 🡪  8/9

**4.3.2 Uyaran/Yanıt Dizileri**

Kullanıcı article düzenlerken başlık, açıklama, içerik yazılarını ve resimlerini düzenleyebilir. Article oluştururken ki şartlar burda da geçerlidir. Article silerken sistem silmeden hemen önce yanlışıkla silinmeleri engellemek için uyarı verir.

**4.3.3 İşlevsel Gereklilikler**

* Kullanıcının girdiği başlık eşsiz ve 10 karakterden fazla olmalıdır. Sistem bu işlemlerin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının girdiği açıklama 10 karakterden fazla olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının girdiği içerik 20 karakterden fazla olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Başlık, açıklama ve içerik yazıları gereklidir boş bırakılamaz. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Sistem üzerinde article oluşturabilmek için giriş yapılmış olması gerekmektedir. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcının yüklediği resmin dosya formatı resim formatları olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Yanlışıkla silinmeleri engellemek için sistem silmeden hemen önce uyarı verir.

### Profil Düzenleme

**4.4.1 Tanım ve Öncelik**

Bu özellik sistemde kullanıcının profilini düzenleyebilmesini sağlar. Sistemin işleyişini doğrudan etkilemediğinden orta öncelikli bir özelliktir.

Fayda 🡪  6/9

Maliyet 🡪  8/9

Risk 🡪  6/9

**4.4.2 Uyaran/Yanıt Dizileri**

Kullanıcı profilini düzenlerken fotoğraf yükleyebilir, fotoğrafını değiştirebilir ve şifresini değiştirebilir. Fotoğraflar resim formatında olmalıdır. Sistem bunun kontrolünü gerçekleştirip buna göre uyarı verir.

**4.4.3 İşlevsel Gereklilikler**

* Yüklenen fotoğraflar resim formatında olmalıdır. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.
* Kullanıcı şifre değiştirirken şifre 6 karakterden az olamaz. Sistem bu işlemin kontrolünü gerçekleştirip kullanıcıyı buna göre uyarmalıdır.

### Article Beğenme

**4.5.1 Tanım ve Öncelik**

Bu özellik sistemde kullanıcının articleları beğenmesini ve dilerse bu beğeniyi geri almasını sağlar. Sistemin işleyişini etkilemediğinden düşük öncelikli bir özelliktir.

Fayda 🡪  5/9

Maliyet 🡪  8/9

Risk 🡪  1/9

**4.5.2 Uyaran/Yanıt Dizileri**

Kullanıcı article beğenirken sisteme giriş yapmış olması gerekmektedir. Sistem bunun kontrolünü gerçekleştirip eğer sisteme giriş yapılmamışsa beğen butonunu deaktif etmelidir. Kullanıcı article beğenmek için beğen butonuna basar ve article beğeni sayısı 1 artar. Beğeniyi geri almak içinse aynı butona tekrar basar ve beğeni sayısı 1 azalır.

**4.5.3 İşlevsel Gereklilikler**

* Kullanıcı article beğenebilmesi için sisteme giriş yapmış olması gerekmektedir. Sistem bu işlemin kontrolünü gerçekleştirip giriş yapmamış kullanıcılar için beğen butonu deaktif olmalıdır.
* Kullanıcı article beğenmek için beğen butonuna basar ve beğeni sayısı 1 artar.
* Kullanıcı dilerse aynı butona basarak beğeniyi geri alabilir ve beğeni sayısı 1 azalır.

### Yorum Yazma

**4.6.1 Tanım ve Öncelik**

Bu özellik sistemde kullanıcının articlelara yorum yapmasını sağlar. Sistemin işleyişini etkilemediğinden düşük öncelikli bir özelliktir.

Fayda 🡪  5/9

Maliyet 🡪  8/9

Risk 🡪  2/9

**4.6.2 Uyaran/Yanıt Dizileri**

Kullanıcı article’a yorum yaparken sisteme giriş yapmış olması gerekmektedir. Sistem bunun kontrolünü gerçekleştirip eğer sisteme giriş yapılmamışsa yorum yapma kısmını deaktif etmelidir. Kullanıcı yorum yapmak için yorum kısmına yazı yazar ve paylaşır böylece article’ın yorum kısmında gözükür.

**4.6.3 İşlevsel Gereklilikler**

* Kullanıcı yorum yapabilmek için sisteme giriş yapmış olması gerekmektedir. Sistem bu işlemin kontrolünü gerçekleştirip giriş yapmamış kullanıcılar için yorum yapma kısmını deaktif etmelidir.
* Kullanıcı yorum yapmak için yorum kısmına yazı yazmalıdır. Bu yorum article’ın yorum kısmında gözükmelidir.

### Article Görüntüleme ve Listeleme

**4.7.1 Tanım ve Öncelik**

Bu özellik sistemde genel olarak article listeleme ve görüntüleme işlemlerini gerçekleştirir. Bu işlemler 3’e ayrılır. Bu işlemlerden birincisi anasayfa içeriklerini yüklemektir. İkincisi kullanıcı articlelarını listelemektir. Üçüncüsü de kullanıcının arama sonucuna göre articleları listeler. Bu özellik sistemin işleyişini doğrudan etkilemediğinden orta öncelikli bir özelliktir.

Fayda 🡪  7/9

Maliyet 🡪  8/9

Risk 🡪  3/9

**4.7.2 Uyaran/Yanıt Dizileri**

Anasayfada popüler ve en son yüklenmiş articlelar görüntülenir. Kullanıcı article araması yaparken istediği başlığa göre arama yapar ve anahtar kelimesine göre articlelar anasayfada listelenir. Kullanıcı kendi articlelarını görüntülerken ekranda kaç tane article’ı olduğunu görebilir.

**4.7.3 İşlevsel Gereklilikler**

* Kullanıcı aramak istediği başlıkla ilgili anahtar kelimeyi arama kısmına girer ve bu anahtar kelimeye göre articlelar anasayfada listelenmelidir.
* Kullanıcı kendi articlelarını görüntülerken ekranda kaç article’a sahip olduğunu görebilir.
* Anasayfada popüler ve en son yazılmış articlelar listelenmelidir.

### Articleları Okuma Listesine Ekleme ve Görüntüleme

**4.8.1 Tanım ve Öncelik**

Bu özellik sistemde kullanıcının articleları okuma listesine ekleyip bu listede görüntüleyebilmesini sağlar. Sistemin işleyişini etkilemediği için düşük öncelikli bir özelliktir.

Fayda 🡪  6/9

Maliyet 🡪  8/9

Risk 🡪  1/9

**4.8.2 Uyaran/Yanıt Dizileri**

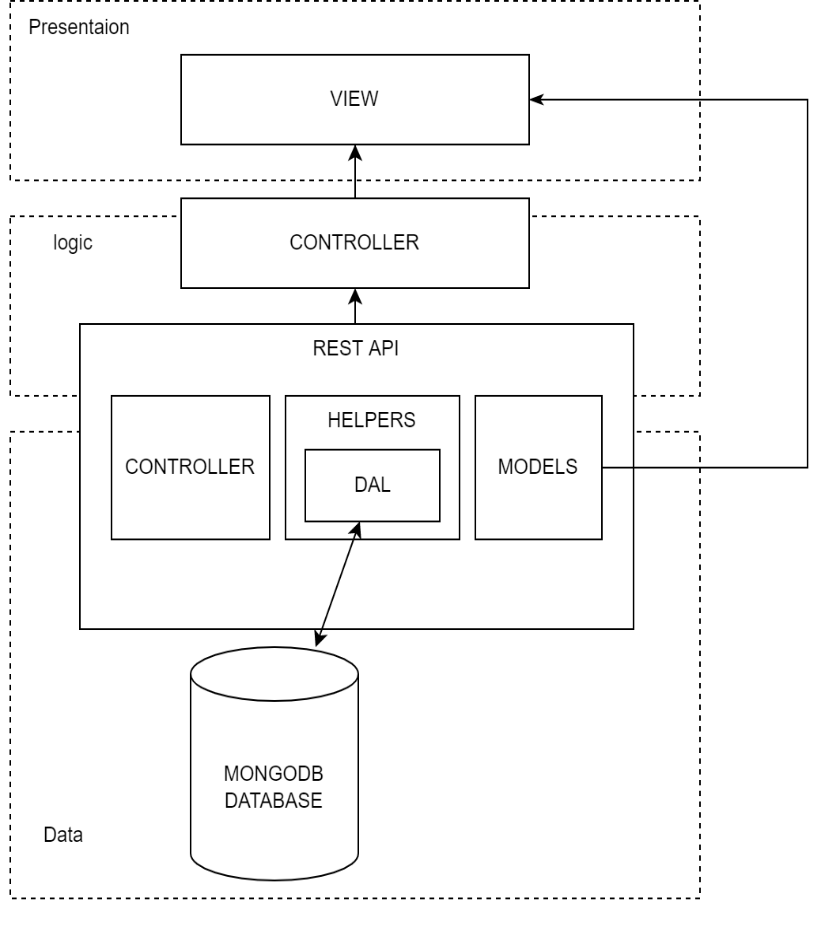
Kullanıcı article’ı okuma listesine ekleyebilmek ve bu articleları görüntüleyebilmek için giriş yapmış olması gerekir ve article’ın ona ait olmaması gerekir. Sistem bunların kontrolünü gerçekletirip buna göre listeye ekleme butonunu deaktif eder.

**4.8.3 İşlevsel Gereklilikler**

* Kullanıcı okuma listesine eklemek istediği article’ın altındaki bookmark butonuna basıp ekleyebilir. Bu işlemi gerçekleştirebilmek için sisteme giriş yapmış olması ve article’ın ona ait olmaması gerekir. Sistem bu işlemlerin kontrolünü gerçekleştirip buna göre bookmark butonunu deaktif hale getirir.
* Kullanıcı okuma listesini görüntüleyebilmek için sisteme giriş yapmış olması gerekmektedir. Sistem bu işlemin kontrolünü gerçekleştirip buna göre giriş yapma ekranına yönlendirir.

## 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.

Bu sistemde yanda verilen diyagramda görüldüğü gibi MVC yapısı kullanılmıştır. Sistem 2 temel nodejs uygulamasından oluşmaktadır. Bunlar API ve Client olarak adlandırılmaktadır. API yandaki diyagramda REST API olarak adlandırılmış olup bileşenleri belirtilmiştir.

API temel olarak backend işlemlerini gerçekleştirir ve sistemin veri tabanıyla bağlantısını sağlar.

Client temel olarak API ile iletişimi gerçekleştirir ve bu doğrultuda kullanıcı arayüzünün oluşturup CRUD operasyonlarının gerçekleştirilmesini sağlar.

Client ve API arasındaki bağlantı http requestleriyle gerçekleştirilir bu bağlamda bir npm paketi olan axiostan yararlanılmıştır. Kullanıcı client tarafına isteklerini iletir ve client tarafında bu istekler axios yardımıyla API’a iletilir. API’nın döndüğün sonuca göre kullanıcı arayüzüne bu sonuç yansıtılır ve işlemler gerçekleştirilir.

API ve database arasındaki bağlantı mongoose ile kurulur. Config dosyasında belirtilen mongo uri aracılığıyla verilen mongodb veritabanına bağlanılır. Client tarafından gönderilen istekler doğrultusunda models klasörünün altında bulunan mongo schemalarından yararlanılıp CRUD operasyonları mongoose yardımı ile gerçekleştirilir.

## Sistem Tasarımı

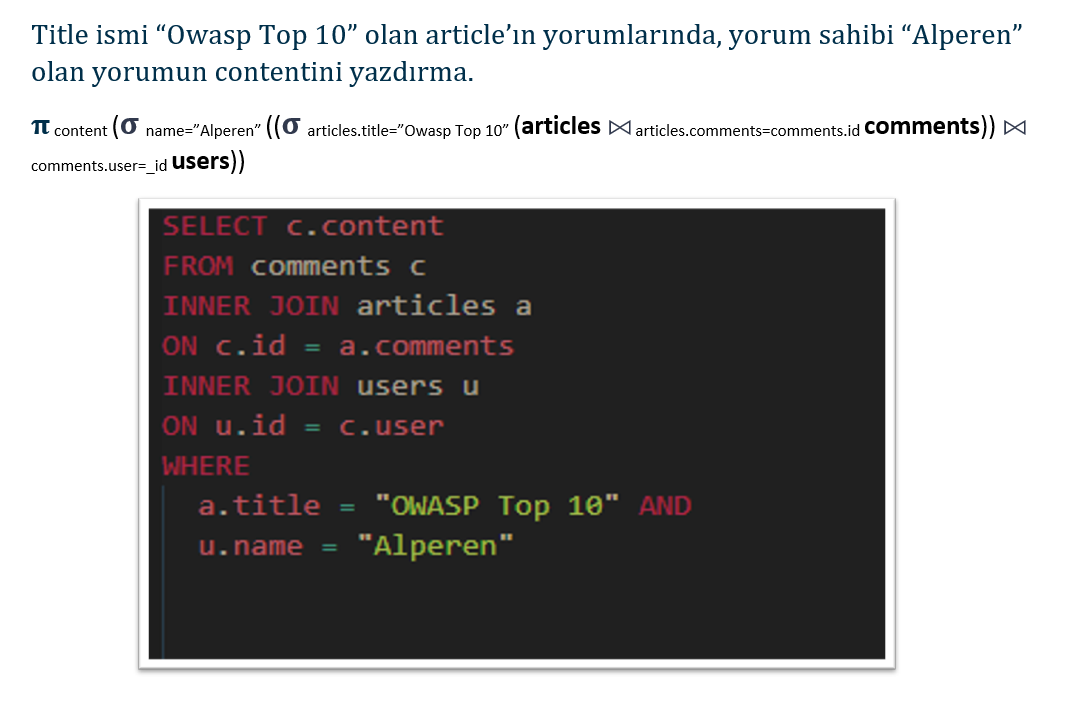
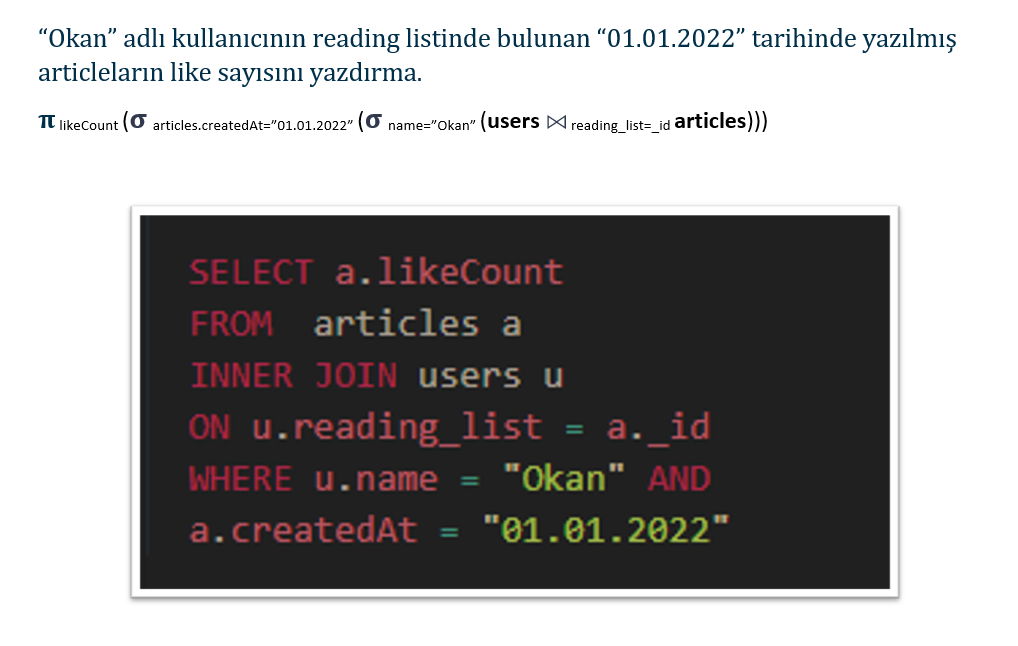
### Varlık-İlişki Diyagramı

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

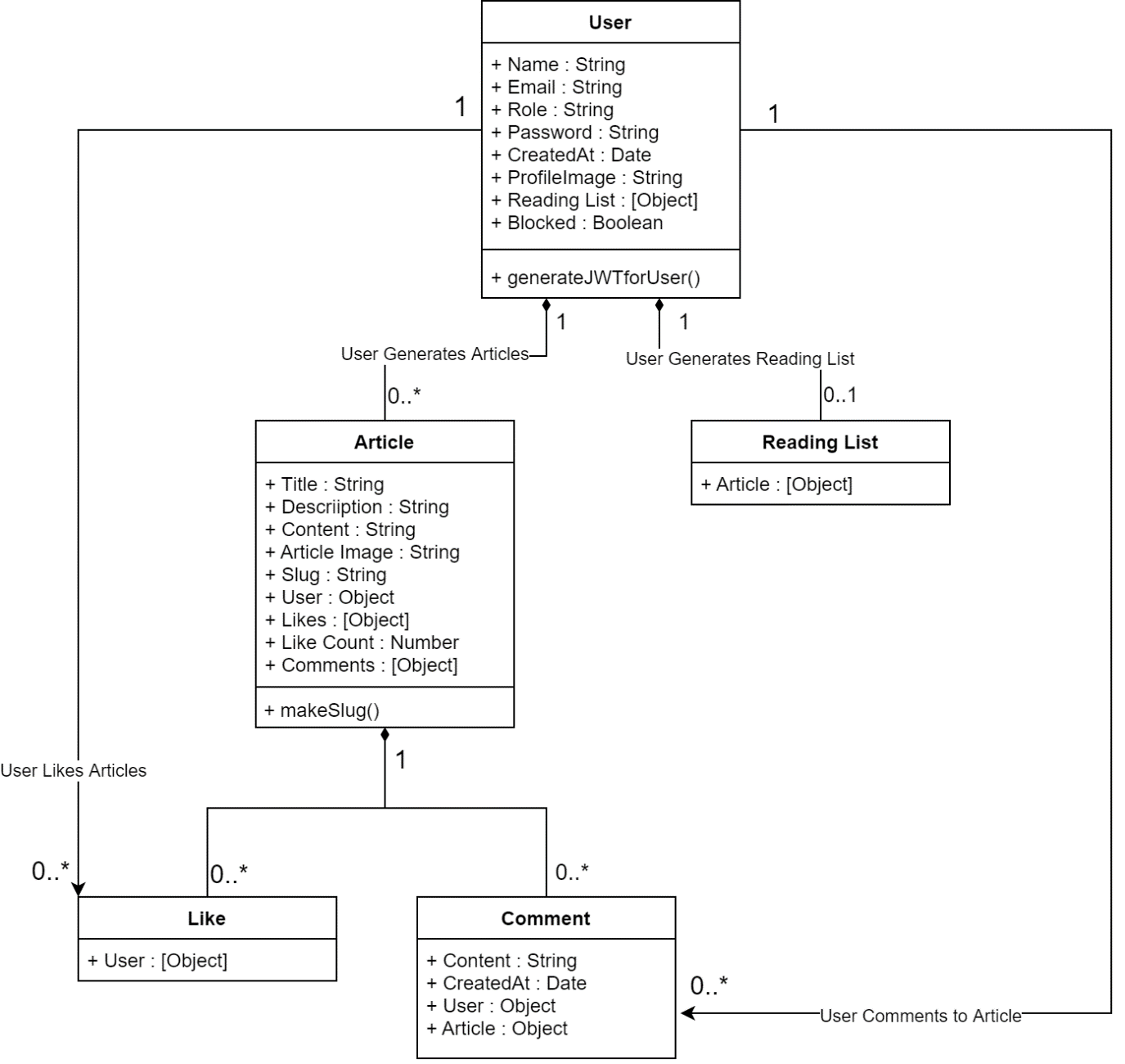
(NoSQL)

### İlişkisel Cebir İfadeleri

*İlişkisel cebir ifadeleri ve onların SQL cümleler*

**

### Sınıf Diyagramı

*UML Class diyagramınızı çizin.*

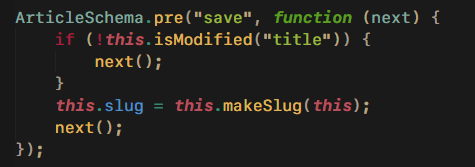
### *CRUD* Matrisi

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** | **CREATE** | **READ** | **UPDATE** | **DELETE** |
| Users | X | X | X | X |
| Comments | X | X |  |  |
| Articles | X | X | X | X |
| Likes | X | X |  | X |
| Reading Lists | X | X |  | X |

### View, Trigger ve Stored Procedure Kullanımı

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

**

Her oluşturulan article için slug oluşturma işlemi.

Trigger 1

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Update edilen her article için eğer article image değişmişse eski article image’i silme işlemi.

Trigger 2



Update işlemi gerçekleştikten sonra eğer title değiştiyse slug’ın güncellenmesi işlemi.

Trigger 3



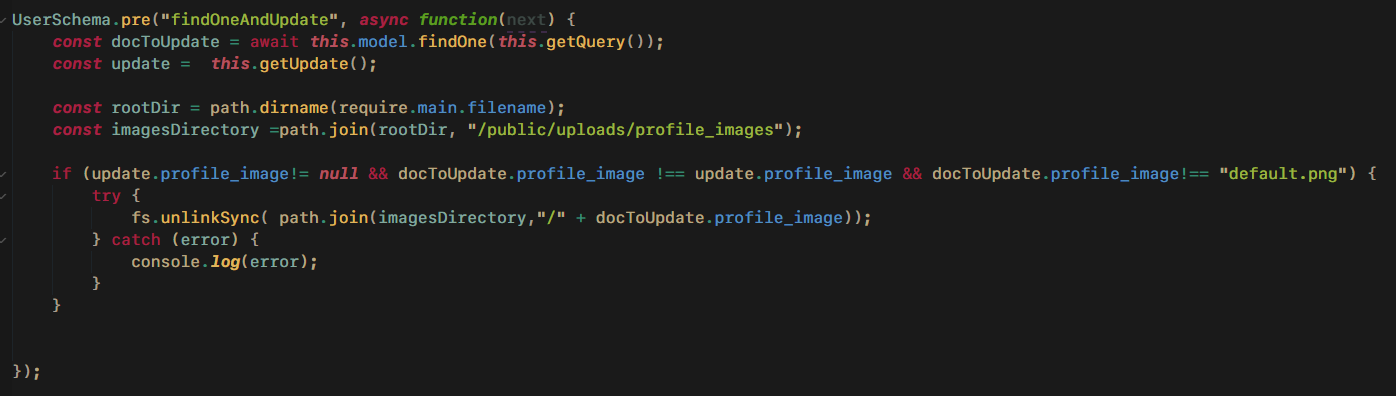
Silinen article’ın commentlerini ve article image’ini silme işlemi.

Trigger 4



User kaydetmeden önce girilen password’un hashlenmesi işlemi.

Trigger 5



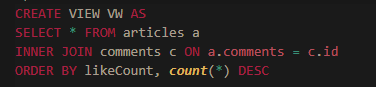
Trigger 6

Update edilen her user için eğer profile image değişmişse eski profile image’i silme işlemi.



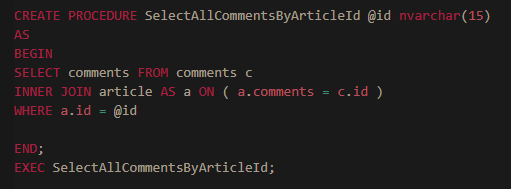
Silinen user’in articlelarını, commentlerini profile image’ini silme işlemi.

Trigger 7



En çok yorumu ve beğeniyi alan articleları listeleme işlemi.

View

**

Stored Procedure

Verilen Articleid’ye göre o article’ın bütün yorumlarını çekme işlemi.

# 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)

***const*** Article = require(**"**d:/yedek win11/small-blog-project/api/models/article**"**)**;**

use small-blog**;**

db**.**createCollection(**"**users**"**)

db**.**createCollection(**"**comments**"**)

db**.**createCollection(**"**articles**"**)

**//**Buradaki veriler collectionların yapısını belirtmek için örnek olarak verilmiştir.

**//**Gerçek deneme verileri json dosyası olarak verilecektir.

**//**Insert işlemleri

db**.**users**.**insert([

    {

        id : 1**,**

        name : **"**sedat**",**

        email : **"**sedat@gmail.com**",**

        role : **"**user**",**

        password : **"**$2a$10$pkyU1bitV65sOmB6Ofdaju8/m6BgLK8dflRuL8Ns1UbAfDY68BZY2**",**

        profile\_image : **"**default.png**",**

        reading\_list : []**,**

        blocked : ***false*,**

        createdAt : **new** Date()

    }**,**

    {

        id : 2**,**

        name : **"**alperen**",**

        email : **"**alperen@gmail.com**",**

        role : **"**user**",**

        password : **"**$2a$10$pkyU1bitV65sOmB6Ofdaju8/m6BgLK8dflRuL8Ns1UbAfDY68BZY2**",**

        profile\_image : **"**default.png**",**

        reading\_list : []**,**

        blocked : ***false*,**

        createdAt : **new** Date()

    }**,**

])

db**.**articles**.**insert([

    {

        id : 1**,**

        title : **"**deneme title**",**

        description : **"**deneme description**",**

        content : **"**deneme content deneme deneme**",**

        article\_image : **"**deneme.png**",**

        user : **"**2kcd23asd3442asd5234**",**

        likes : []**,**

        likeCount : 0**,**

        comments : []**,**

        createdAt : **new** Date()**,**

        slug : **"**deneme-title**"**

    }**,**

    {

        id : 2**,**

        title : **"**deneme 2 title**",**

        description : **"**deneme 2 description**",**

        content : **"**deneme 2 content 2  deneme 2 deneme**",**

        article\_image : **"**deneme2.png**",**

        user : **"**2bdd43afd3542aly5244**",**

        likes : []**,**

        likeCount : 0**,**

        comments : []**,**

        createdAt : **new** Date()**,**

        slug : **"**deneme-2-title**"**

    }**,**

])

db**.**comments**.**insert([

    {

        id : 1**,**

        content : **"**Yazmaya devam etmelisin!**",**

        user : **"**2basdas3423klk2ladf12**",**

        article : **"**23232adfsda3232**",**

        createdAt : **new** Date()

    }**,**

    {

        id : 2**,**

        content : **"**Test comment**",**

        user : **"**2basdas342aspolfkaladf12**",**

        article : **"**23232adfs33232**",**

        createdAt : **new** Date()

      }**,**

])

**//**Update işlemleri

db**.**users**.**update

(

    {

        id : 1

    }**,**

    {

        $set :

        {

**"**name**"** : **"**Utku**",**

**"**profile-image**"** : **"**new\_photo.png**"**

        }

    }

)

db**.**articles**.**update

(

    {

        id : 2

    }**,**

    {

        $set :

        {

**"**title**"** : **"**New article title**",**

**"**slug**"** : **"**New-article-title**"**

        }

    }

)

db**.**comments**.**update

(

    {

        id : 1

    }**,**

    {

        $set :

        {

**"**content**"** : **"**Çok beğendim**"**

        }

    }

)

**//**Delete işlemleri

db**.**users**.**remove( { id: **"**2**"** } )

db**.**articles**.**remove( { id: **"**1**"** } )

db**.**comments**.**remove( { id: **"**2**"** } )

**//** örnek rapor oluşturma sorguları mongoose

**//** belirli tarihler arasında yazılmış tüm articleları like ve paylaşılma tarihine göre sıralanması

Article**.**find({

    created\_on: {

        $gte: **new** Date(2021**,** 7**,** 14)**,**

        $lt: **new** Date(2022**,** 7**,** 15)

    }

})**.**sort(**"**-likeCount -createdAt**"**)**;**

**//** aranan isme sahip kullanıcılar reading listleri

***var*** name**;**

User**.**find({name: { $regex: **'**.\***'** + name + **'**.\***'** } })

**.**populate(**'**reading\_list**'**)

**.**select({name:1**,** reading\_list: 1})**;**

**//** verilen tarihler arasında yazılmış yorumların user ve article bilgisi

Comment**.**find({

    created\_on: {

        $gte: **new** Date(2021**,** 7**,** 14)**,**

        $lt: **new** Date(2022**,** 7**,** 15)

    }

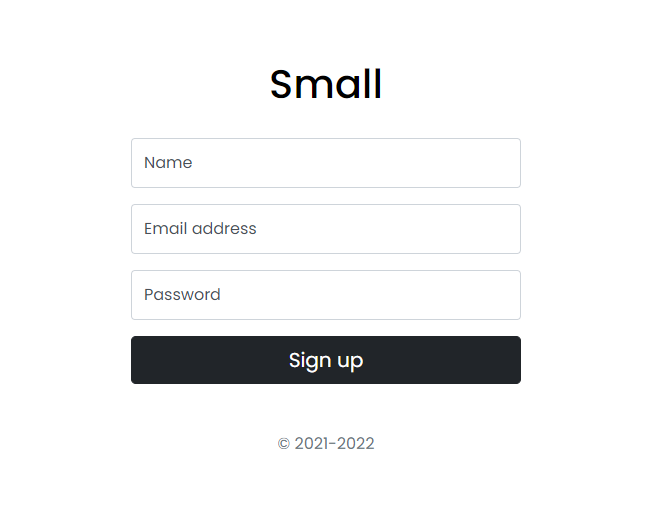
})**.**populate(**'**article**'**)

**.**populate(**'**user**'**)

**.**select({article: 1**,**user : 1})**;**

# Ek B: Ekran Görüntüleri

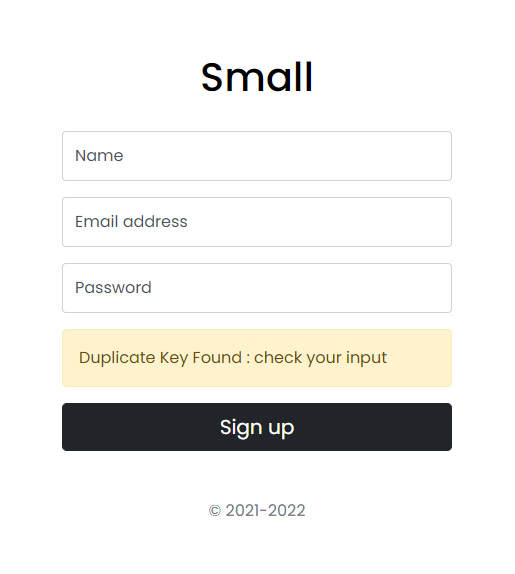
*Ana işlemler için kısa açıklamalı arayüzlerinizin ekran görüntüleri*



Şekil 3 -Register Ekranı

Şekil 2 - Login Ekranı (Doldurulmuş)

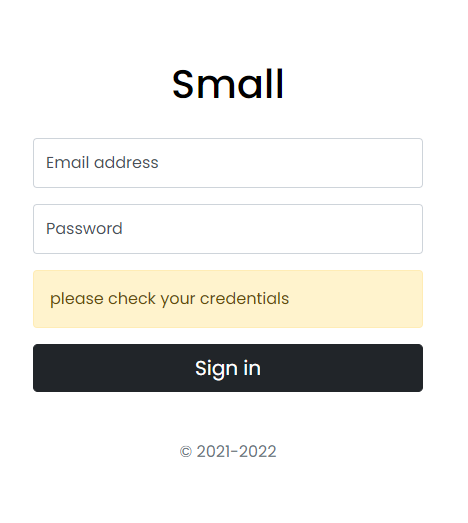
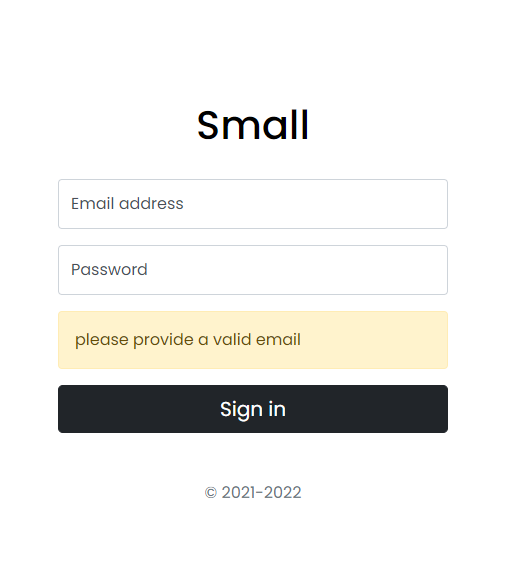
Şekil 4 - Login Ekranı

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 5 - Register Ekranı ( )

Şekil 4 - Register Ekranı (Şifre Uzunluk Validasyonu)

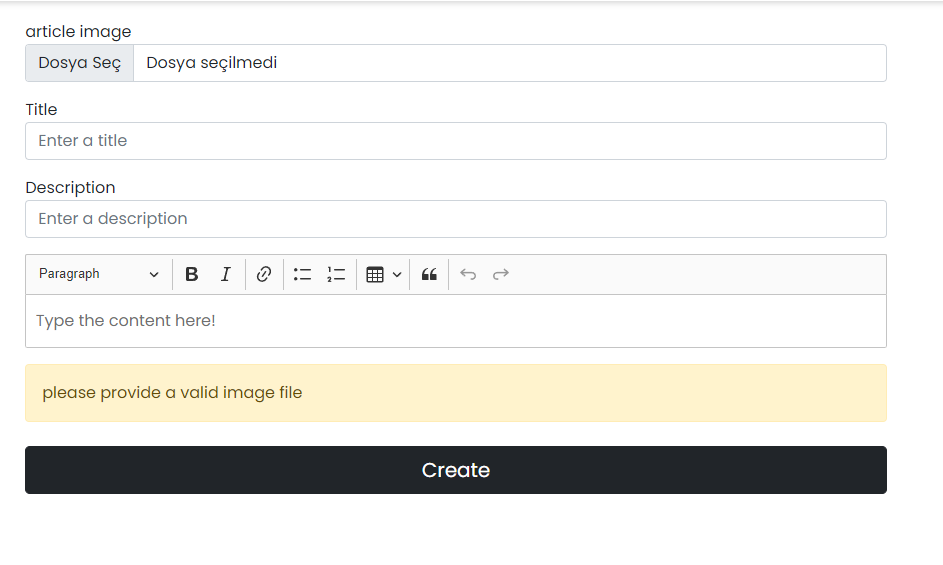


Şekil 7 - Login Ekranı (Mail Validasyonu)

Şekil 6 - Login Ekranı (Giriş Bilgileri Validasyonu)

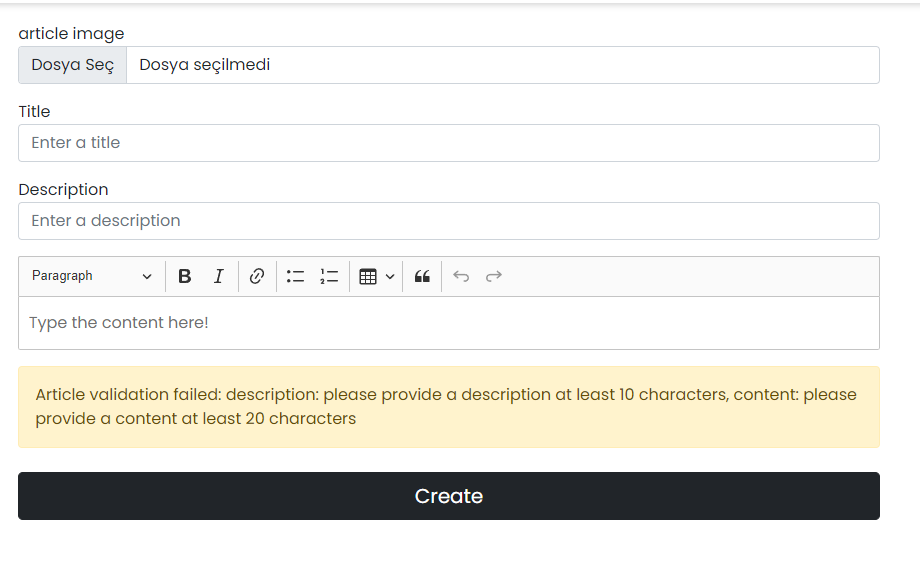
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



Şekil 8 - Blog Oluşturma

Şekil 9 - Blog Oluşturma (Resim Dosyası Validasyon)



Şekil 10 - Blog Oluşturma ( Yazı Uzunluğu Validasyon)

metin içeren bir resim

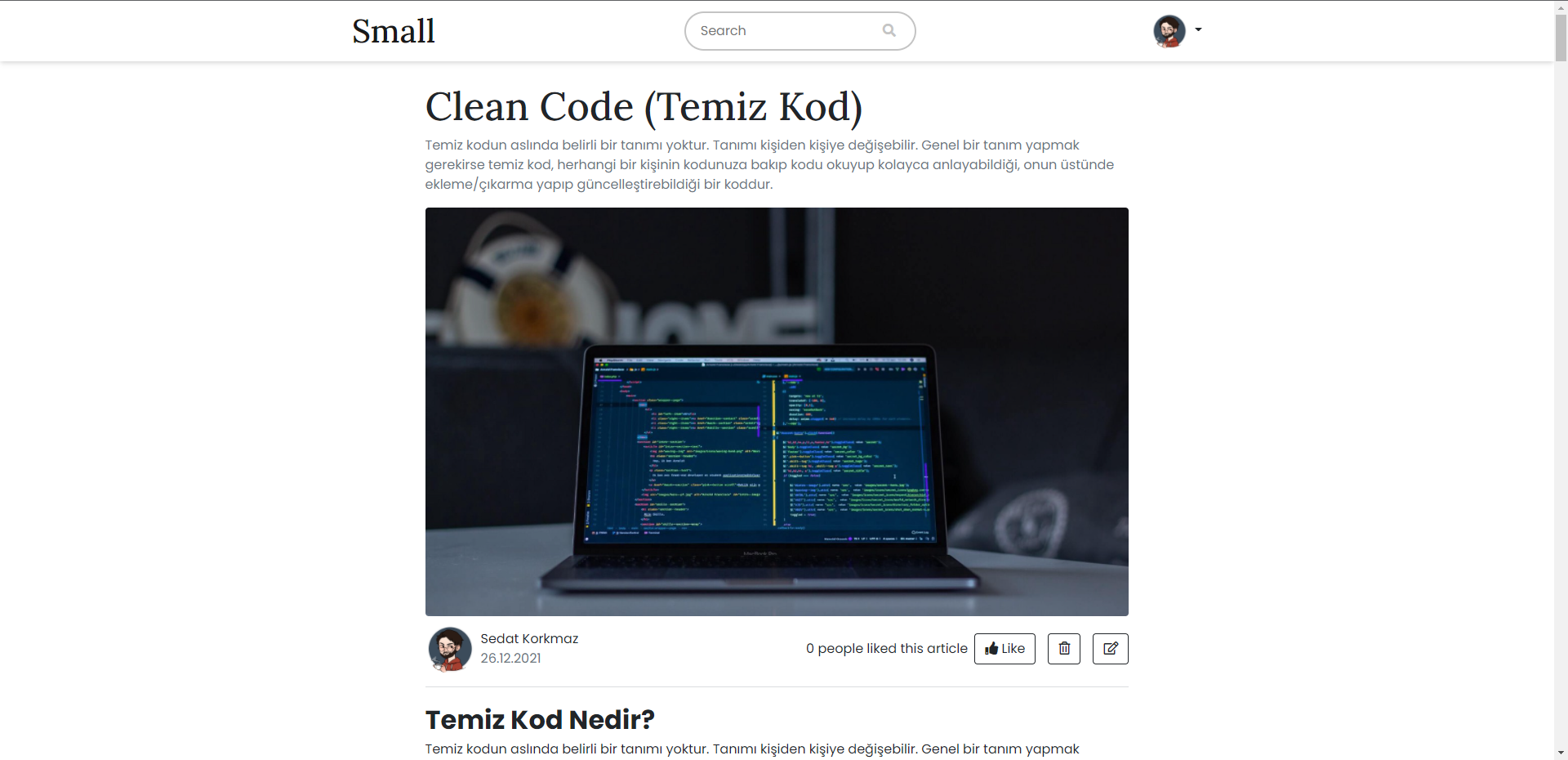
Açıklama otomatik olarak oluşturuldu

Şekil 11 – Yorum Ekranı

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

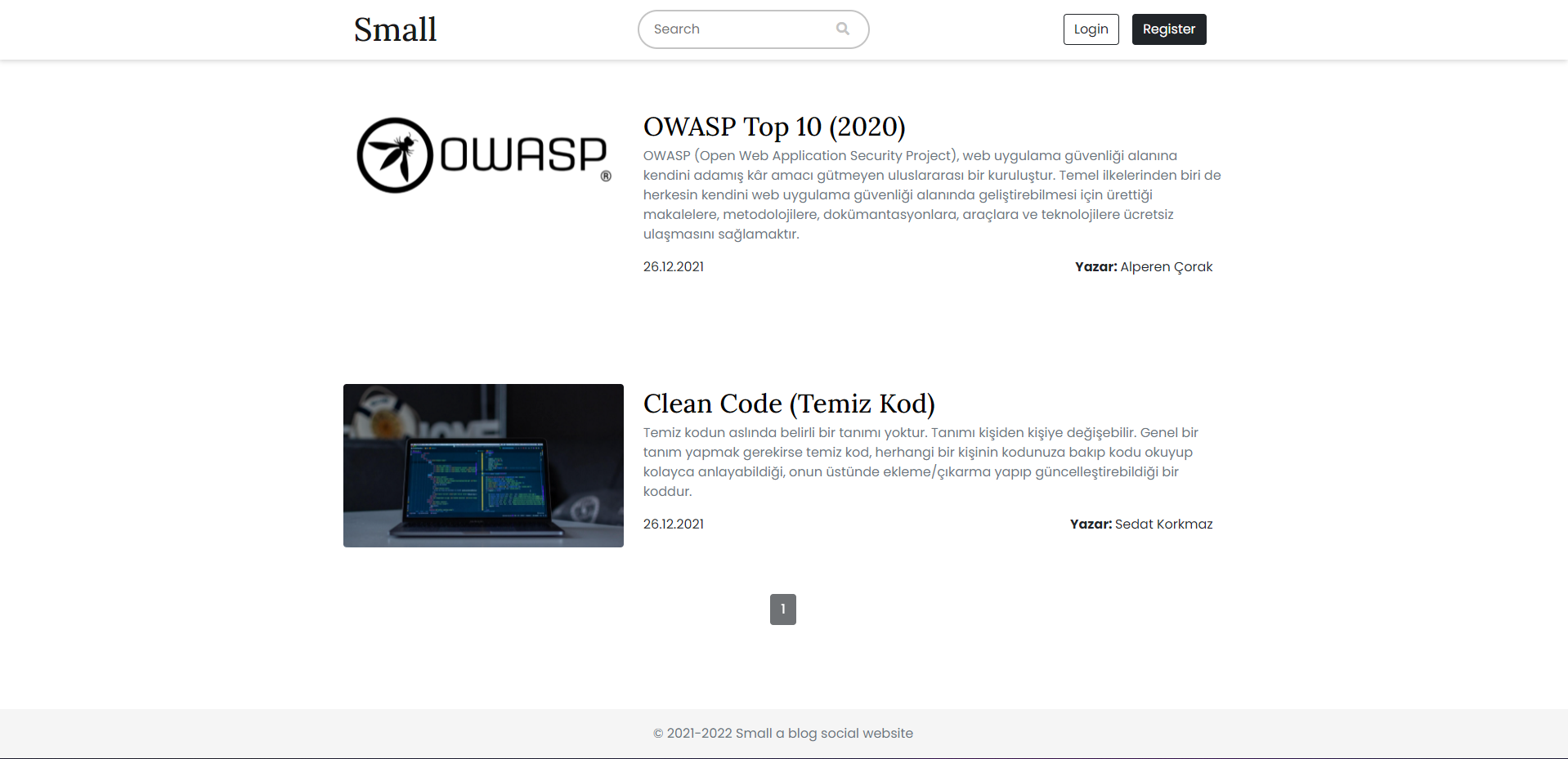
Şekil 12 - Yorum Ekleme



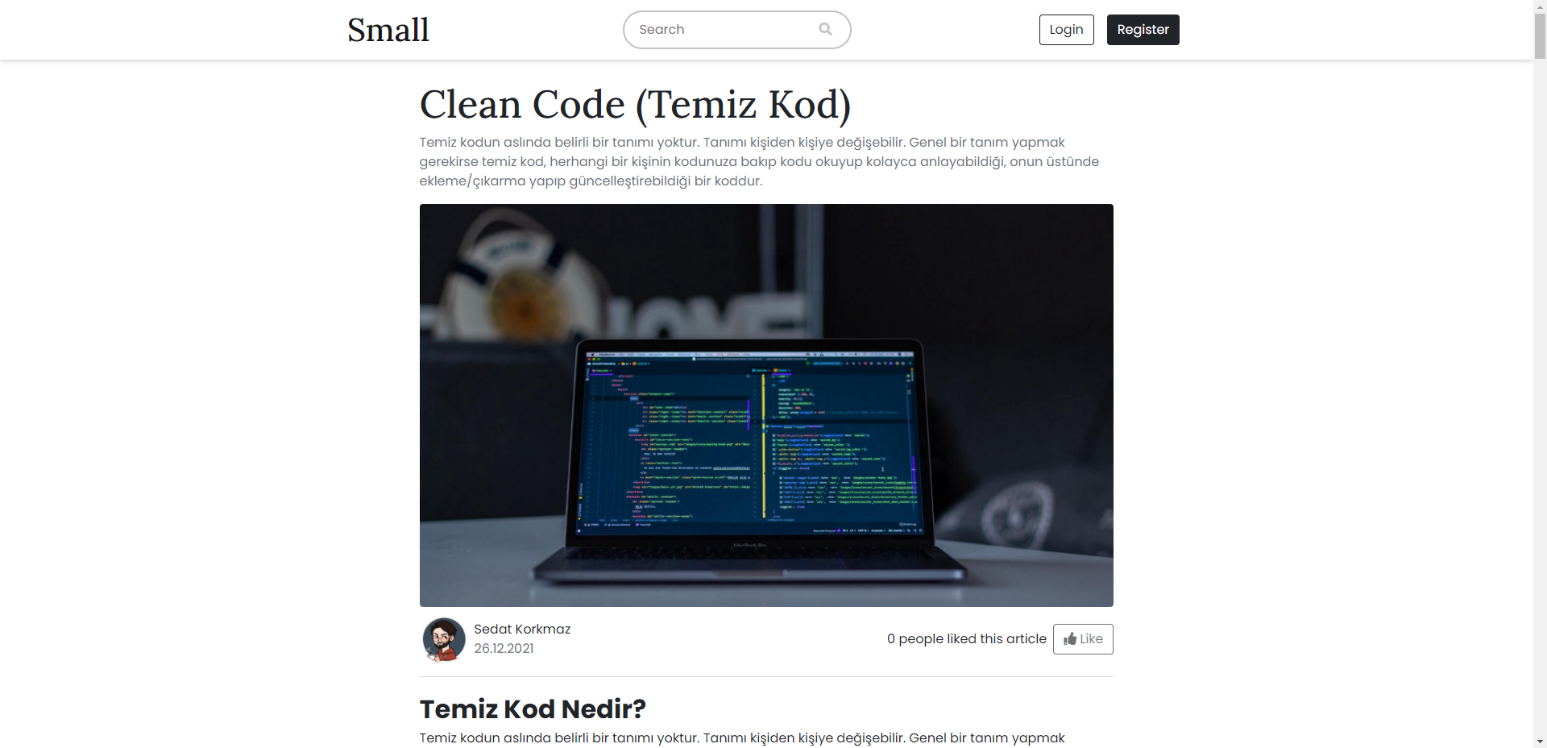
Şekil 13 - Blog Görüntüleme (Yazar)



Şekil 14 - Blog Görüntüleme (Kullanıcı)

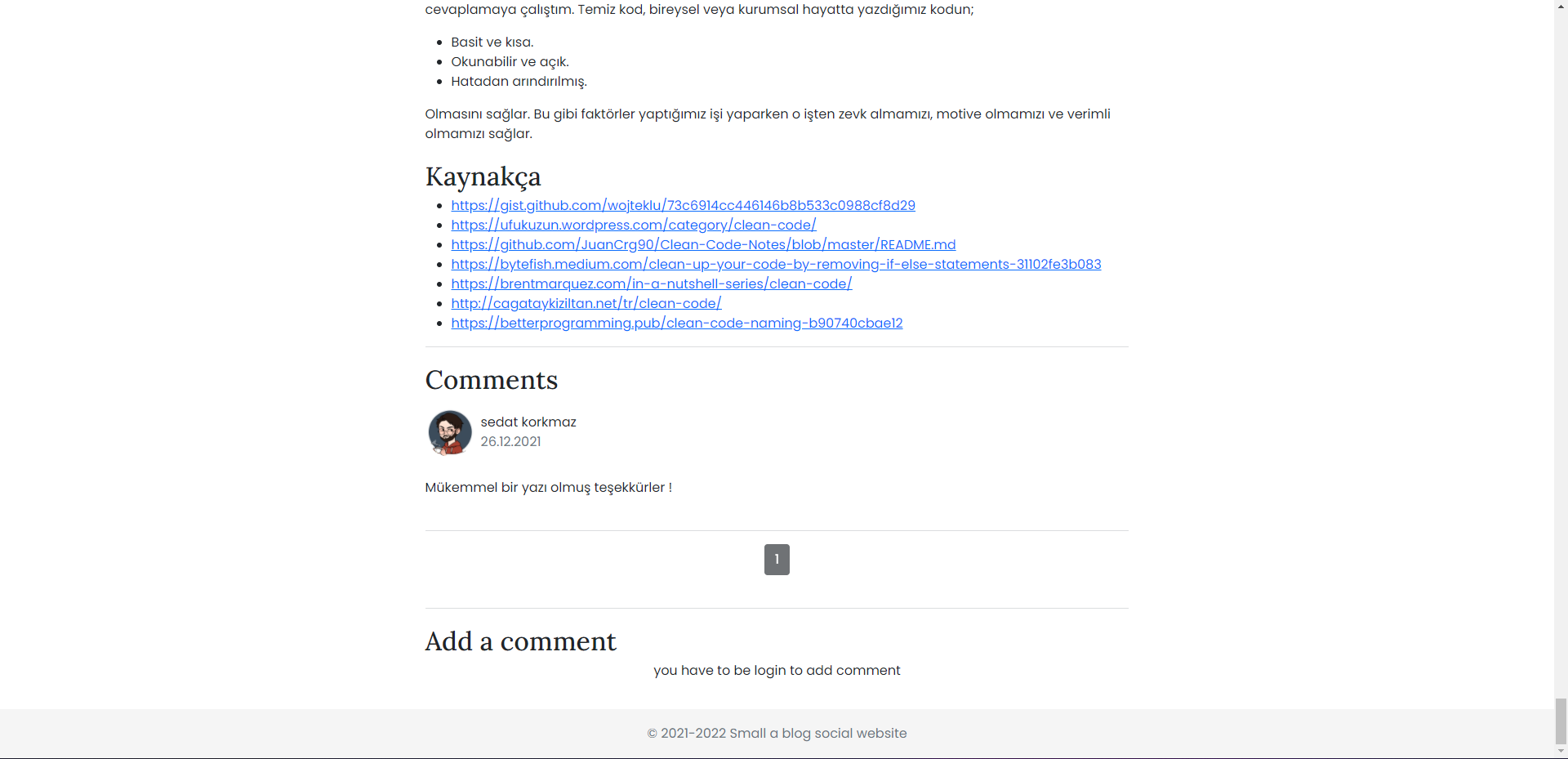
****

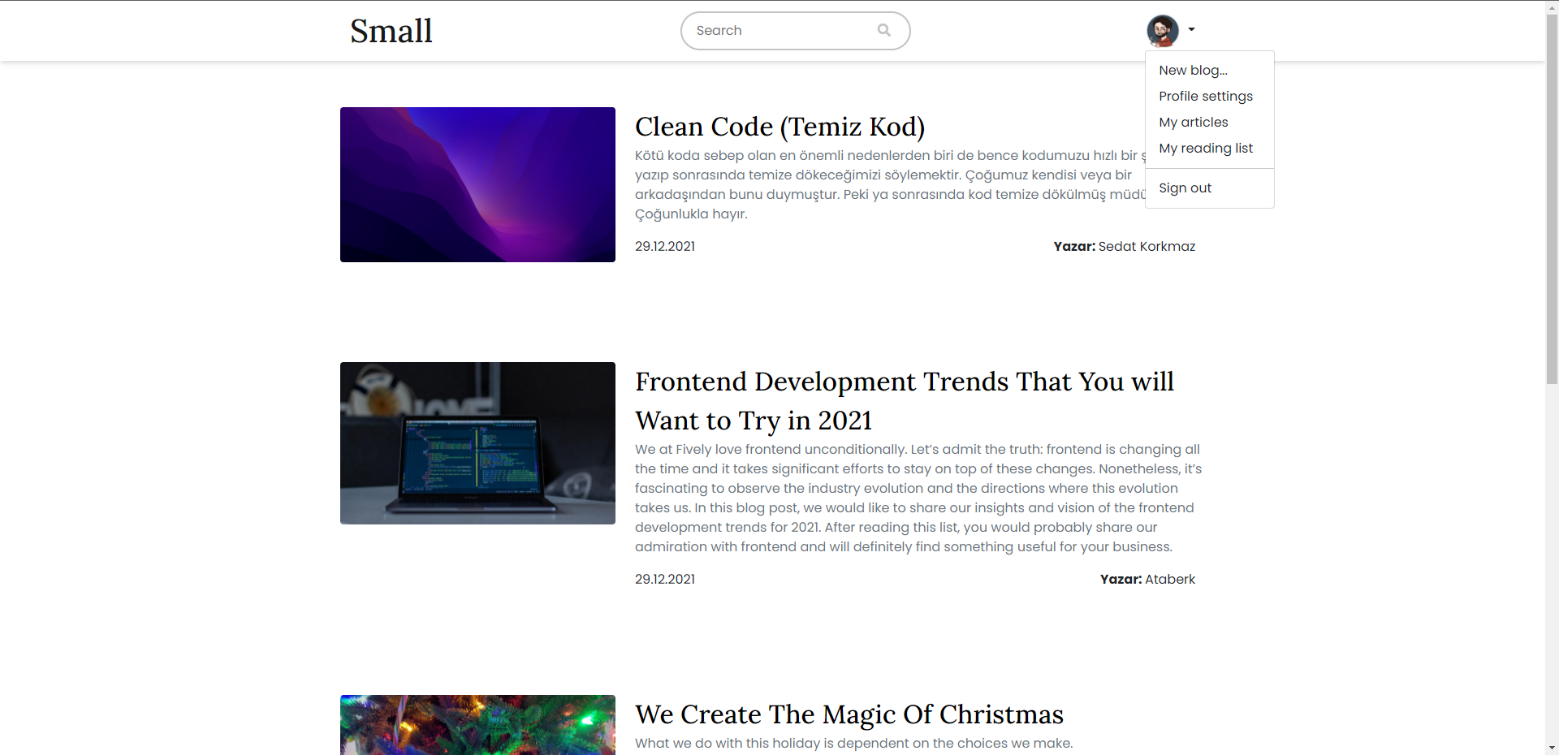
Şekil 15 - Site Ziyaretçisi Anasayfa Ekranı

****

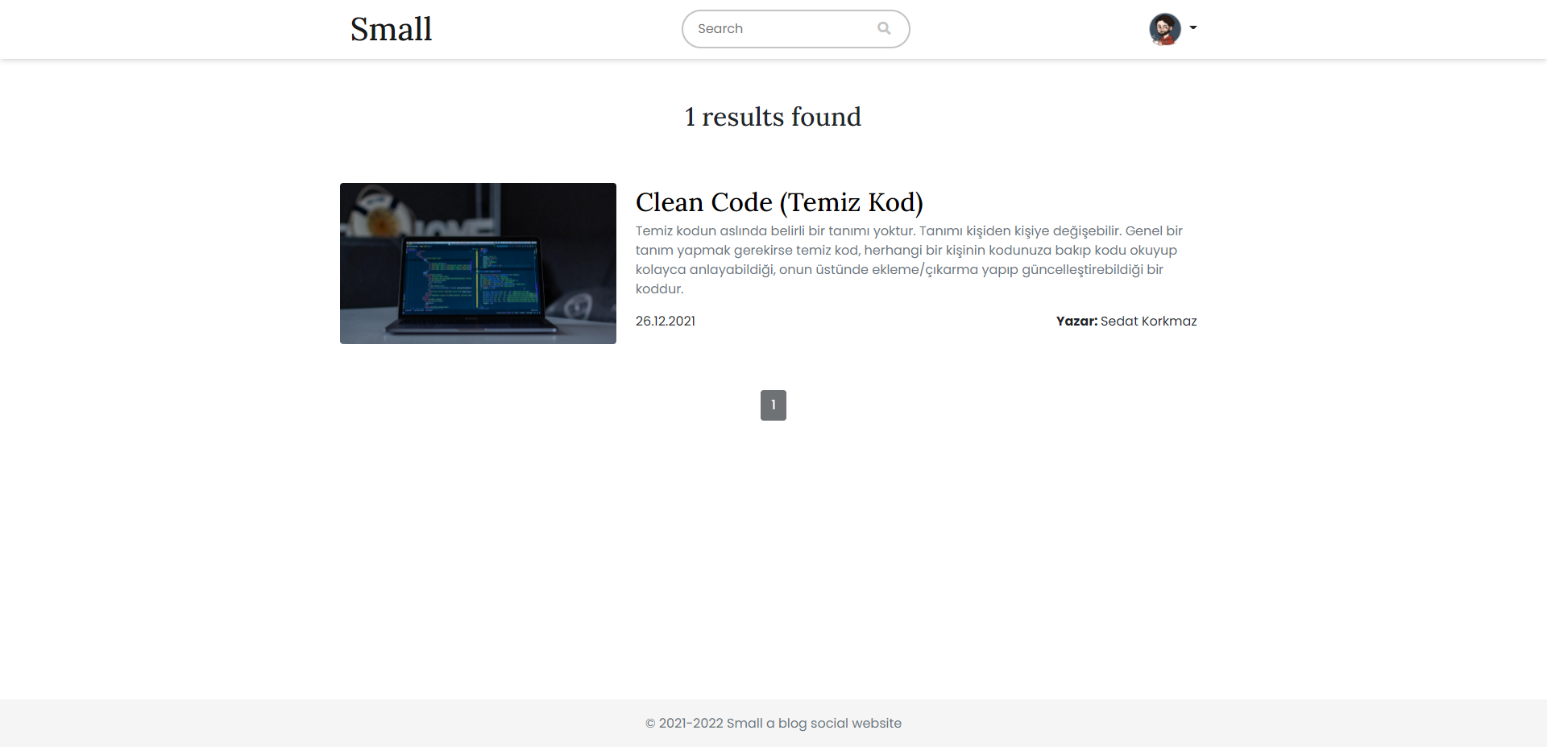
Şekil 17 - Site Ziyaretçisi Yorum Ekranı

Şekil 16 - Site Ziyaretçisi Blog Görüntüleme

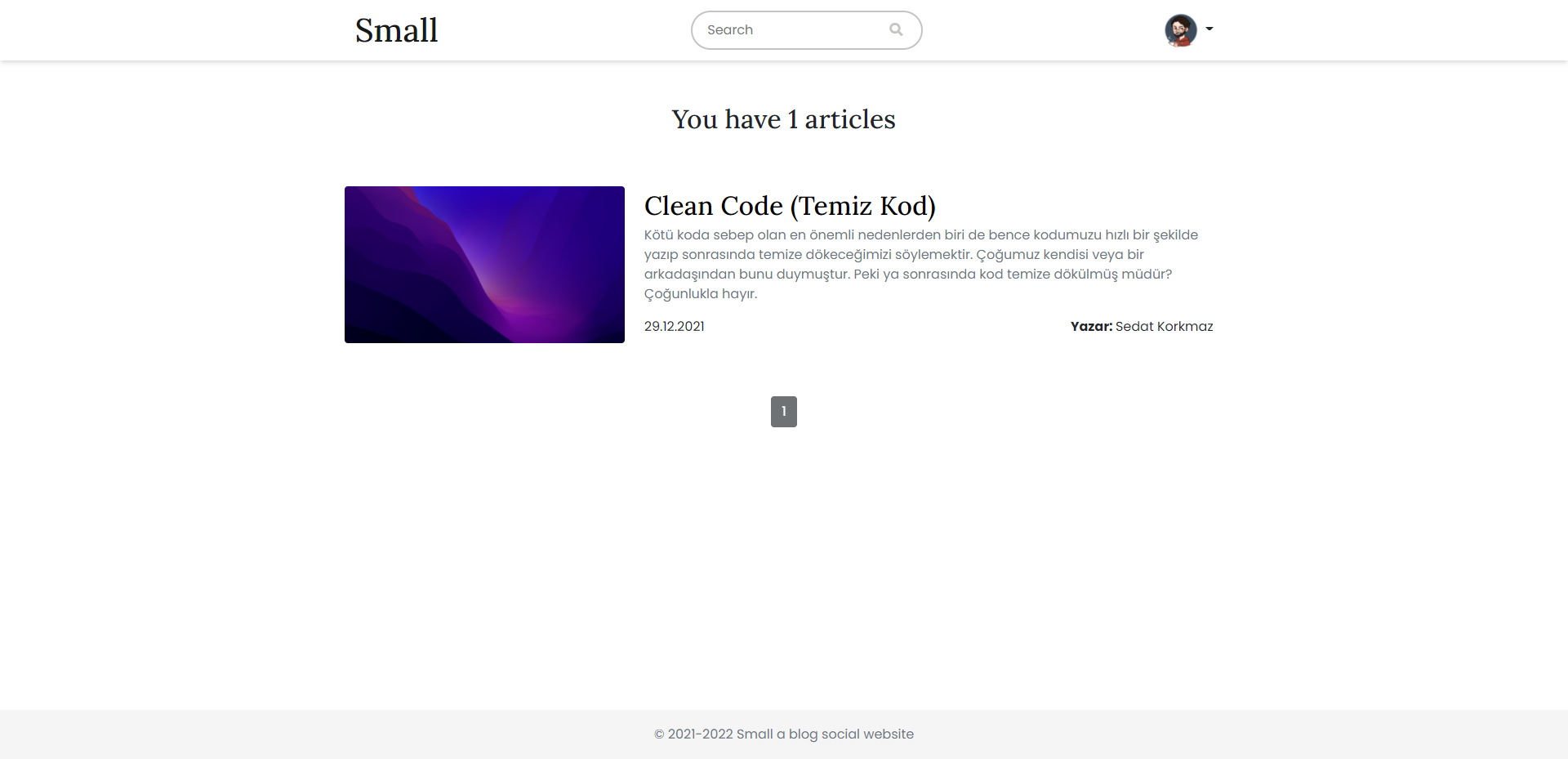
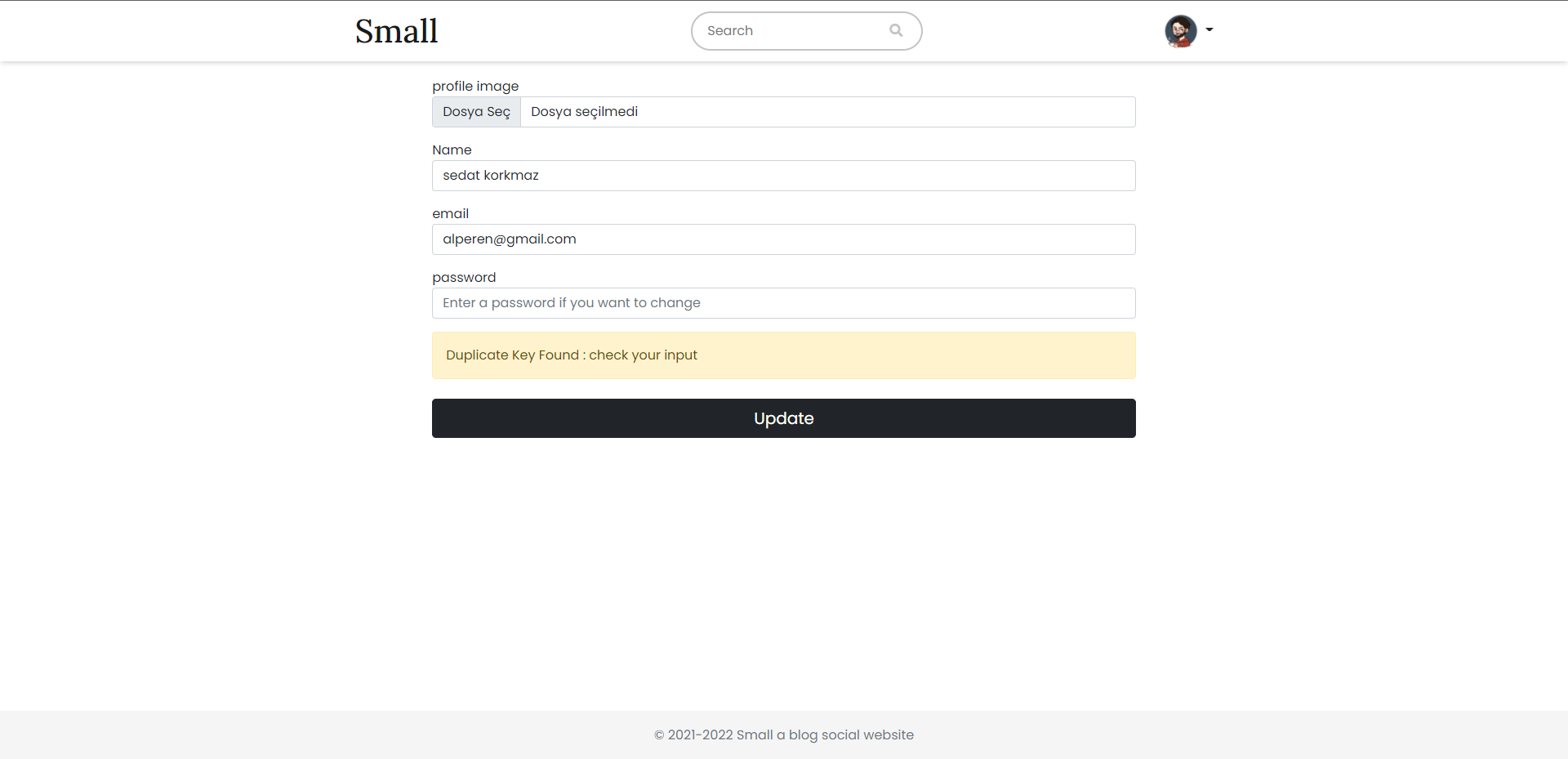
****

****

Şekil 18 - Profil Menüsü

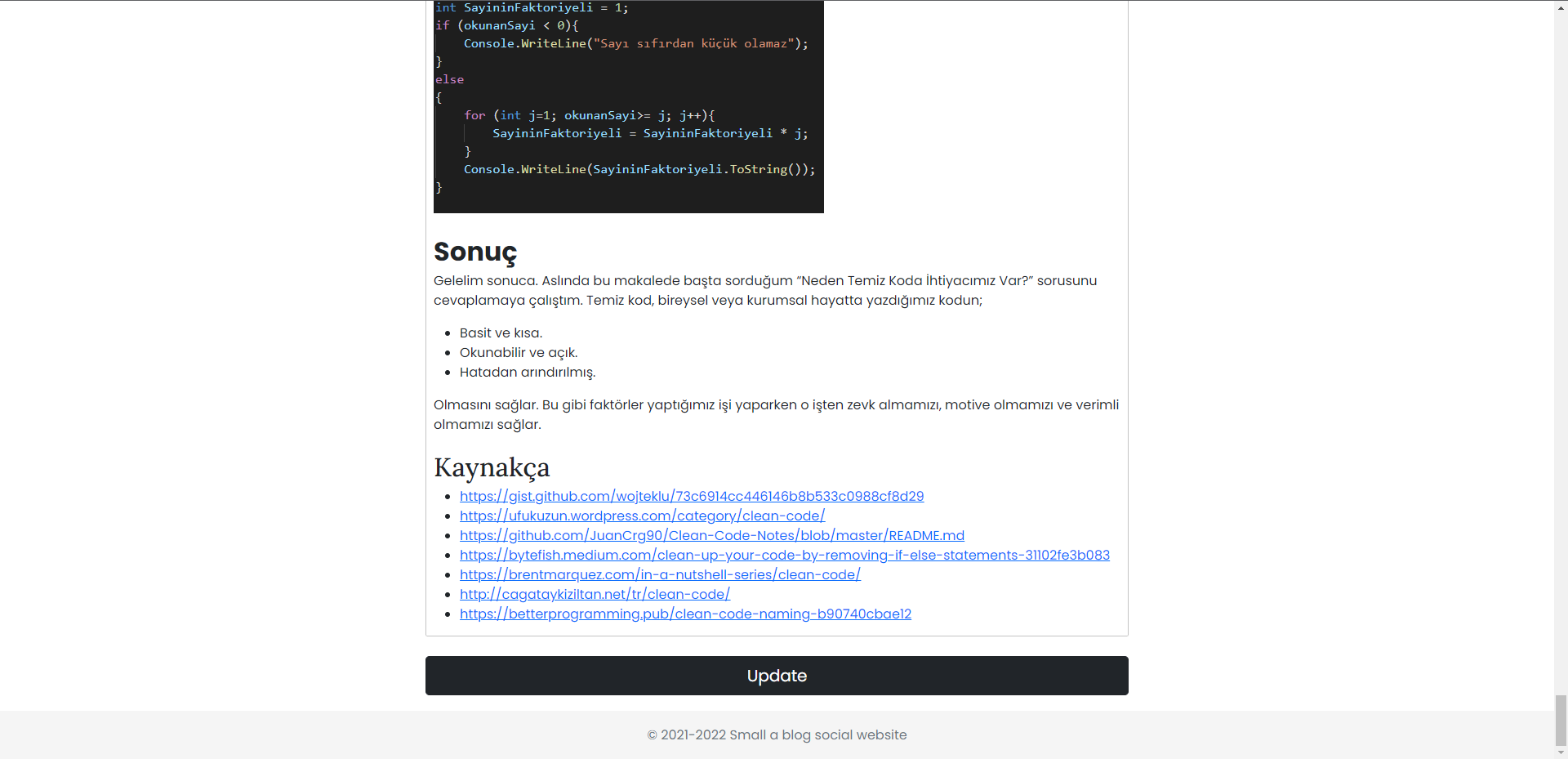
****

Şekil 19 - Blog Arama Sonucu

****

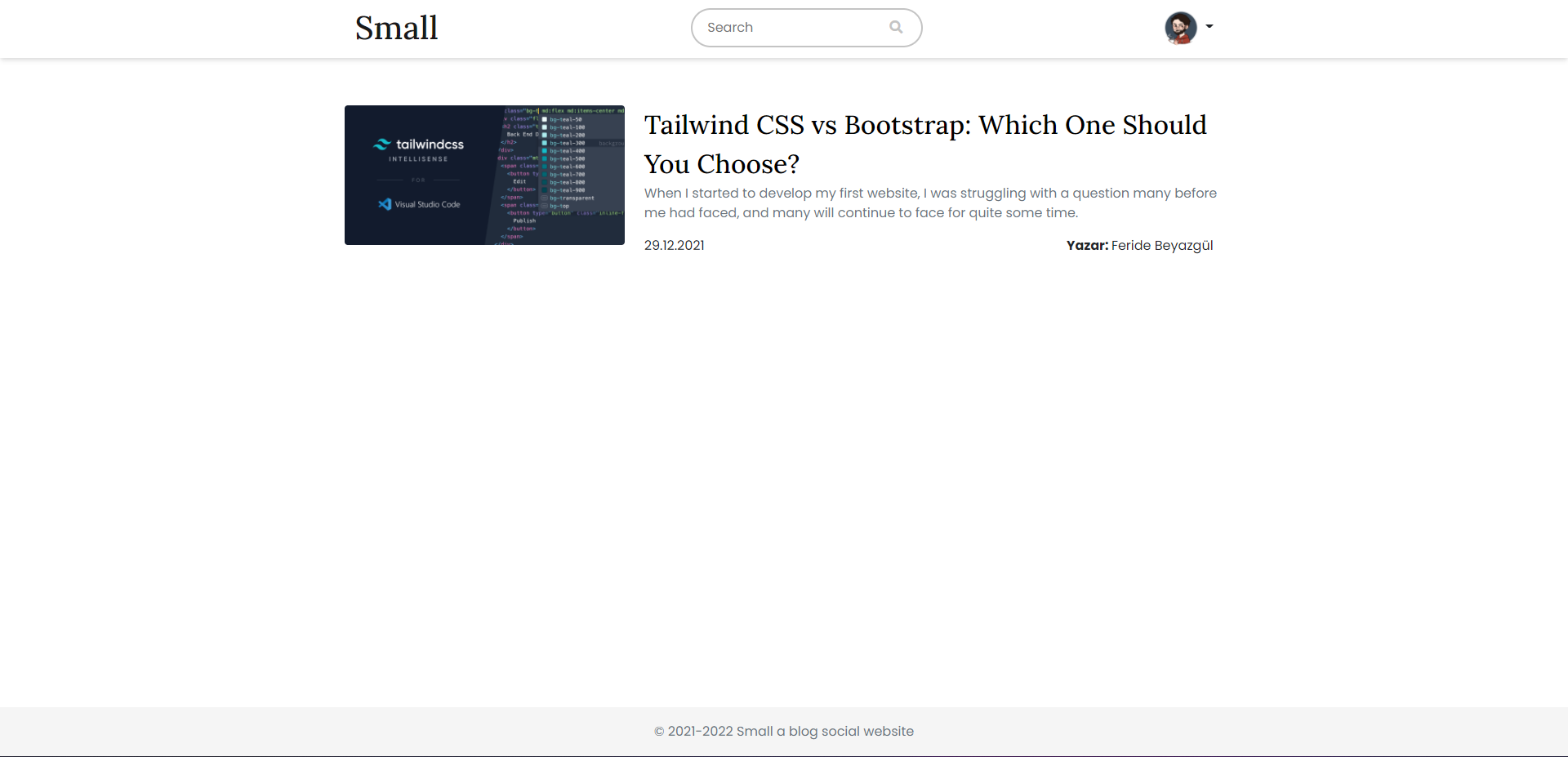
Şekil 20 - Profil Düzenleme Hatası

Şekil 21 - Kendi Yazılarını Görme Ekranı

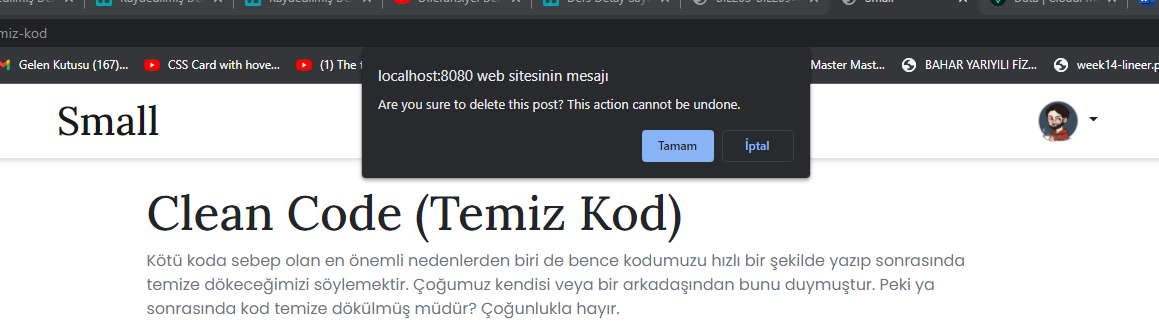


Şekil 22 - Blog Düzenleme - 1

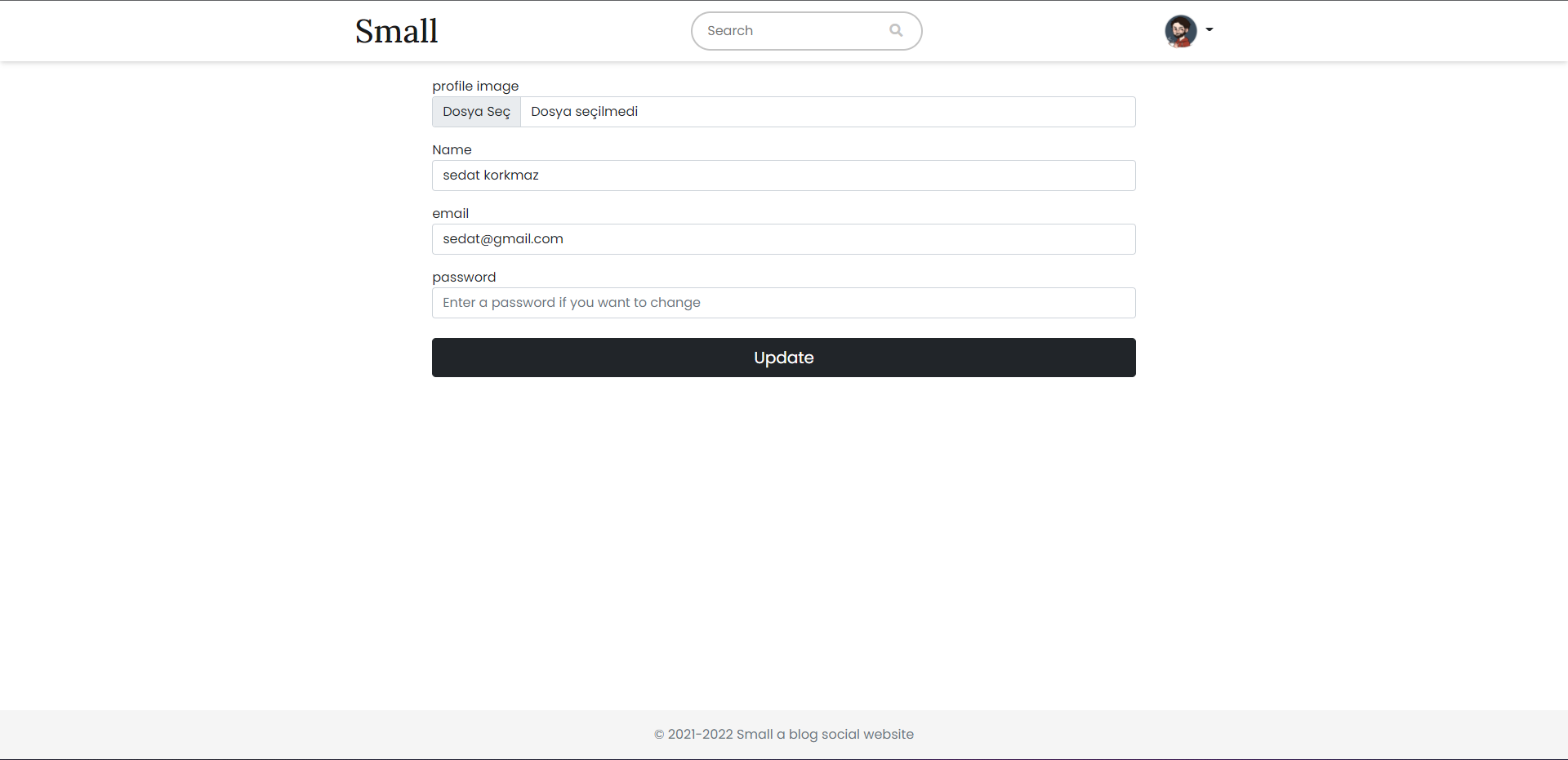
Şekil 23 - Blog Düzenleme - 2



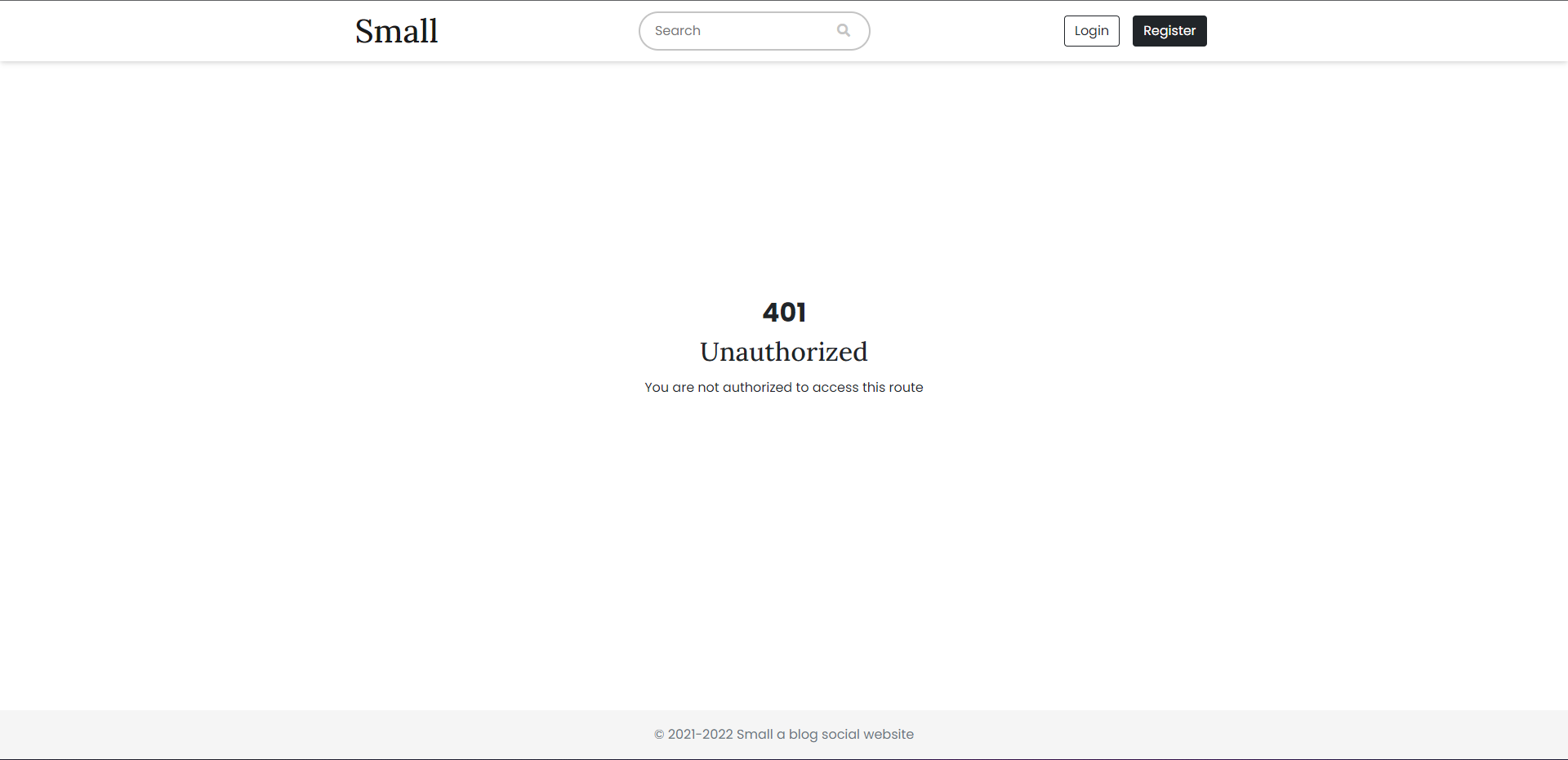
Şekil 24 – Okuma Listesi Ekranı



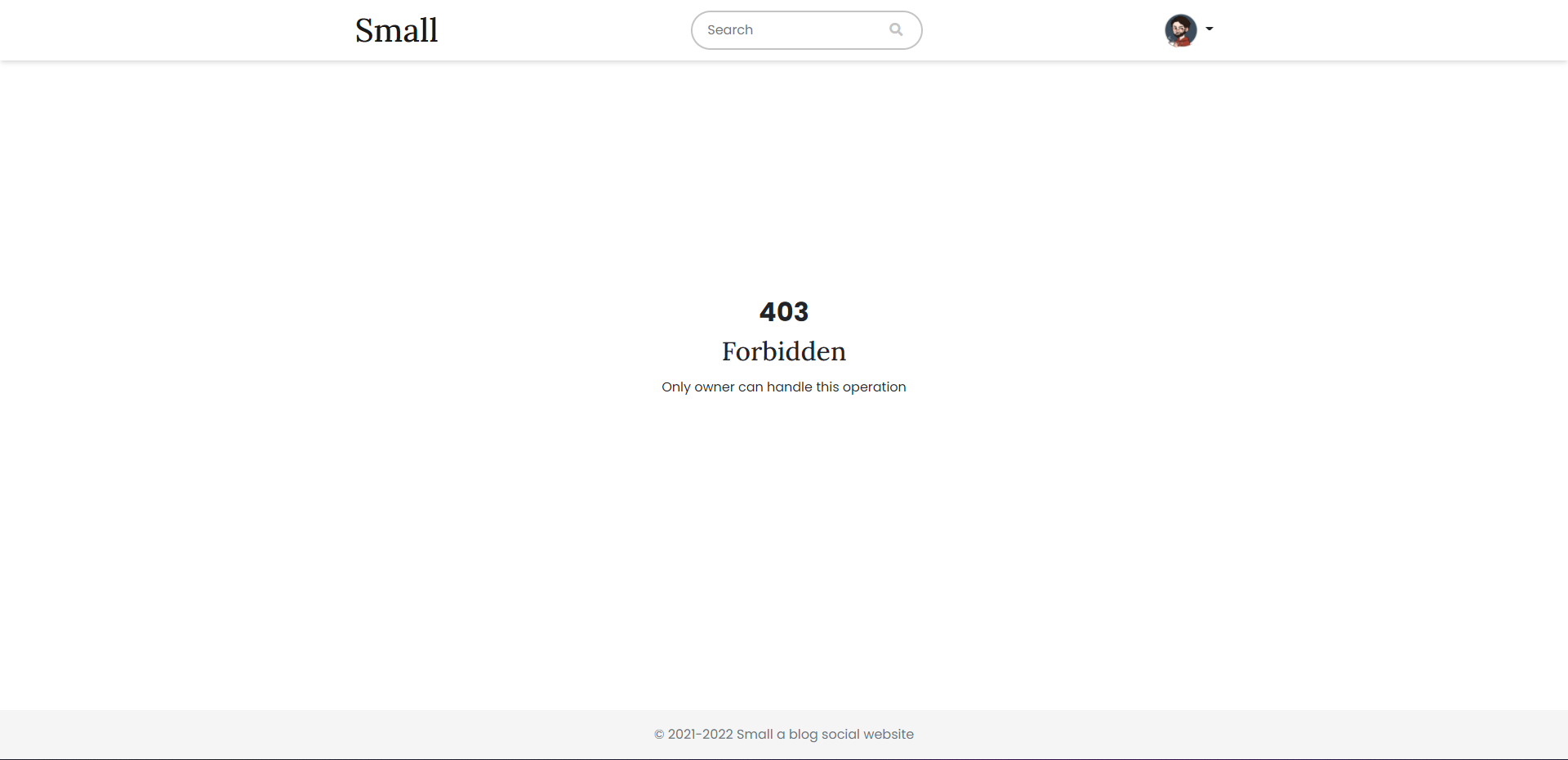
Şekil 25 – Blog Silme (Pop-up uyarısı)



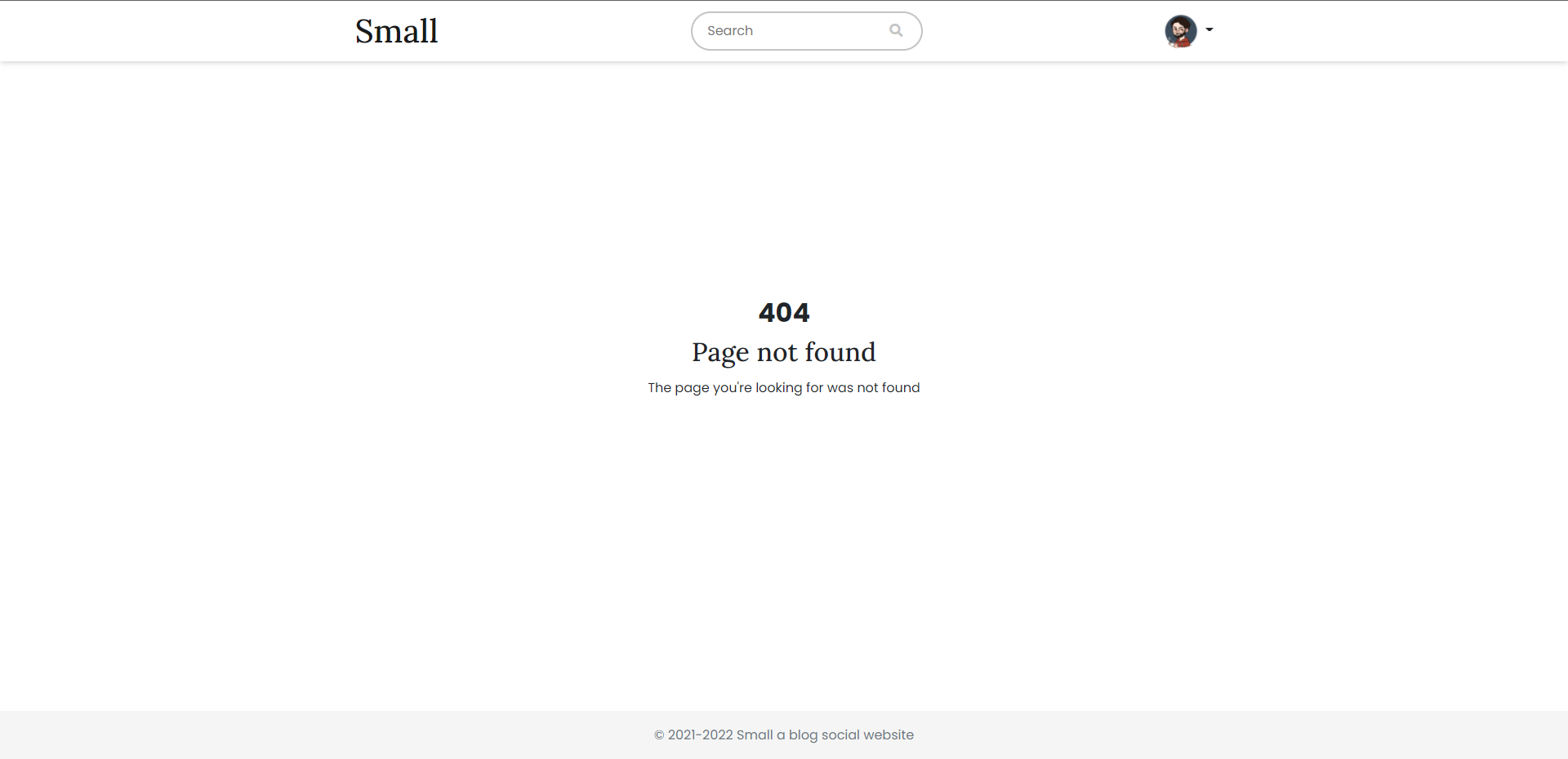
Şekil 26 – Profil Düzenleme



Şekil 27 – Hata 401 (Yetkisiz Erişim)



Şekil 28 – Hata 403 (Yasak)



Şekil 29 – Hata 404 (Sayfa Bulunamadı)

# Ek C: To Be Determined List (TBDL)