

Blog SITE

Serhat KAÇMAZ
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
İstanbul, Türkiye
serhatkacmaz3@gmail.com

181307036

Abstract— The blog site project will be a platform that will allow users to read blogs in multiple categories, make comments, assign likes and dislikes, add the blogs they want to favorites and share.

Keywords—blog, user, like, dislike

I. GİRİŞ

Günümüzde internetin yaygınlaşmasıyla birlikte bloglar, bireylerin ve kurumların kendilerini ifade etmek, bilgi paylaşmak ve etkileşimde bulunmak için en popüler araçlardan biri haline gelmiştir. Bu projede, kullanıcıların birden fazla kategorideki blogları okuyabilmesine, yorum yapabilmesine, beğeni ve beğenmeme ifadesi kullanabilmesine, istedikleri blogları favorilere ekleyebilmesine ve paylaşabilmesine olanak sağlayan bir blog sitesi oluşturmayı hedeflemekteyiz.

Aynı zamanda yöneticilerin de siteye giriş yaparak blogları, kullanıcıları ve diğer işlemleri yönetebilecekleri bir yönetici paneli bulunacaktır. Bu proje, kullanıcıların ilgi duydukları konularda blogları okuyabilecekleri, blog yazarlarıyla etkileşimde bulunabilecekleri ve kendi fikirlerini paylaşabilecekleri bir ortam sunmayı amaçlamaktadır.

Web uygulaması için HTML, CSS, JavaScript ve jQuery gibi yaygın front-end teknolojileri kullanılacak, .Net ve Microsoft SQL Server ise back-end tarafında tercih edilecektir. Veri tabanı oluşturulurken Code First yaklaşımı kullanılacak ve Entity Framework Core ORM teknolojisi entegrasyonu yapılacaktır.

Bu raporda, blog sitesi projesinin geliştirme süreci, kullanılan teknolojiler, proje yapısı ve tasarım kalıpları gibi detaylar ele alınacaktır. Ayrıca projenin amaçları, hedef kitle, beklentiler ve elde edilmesi beklenen sonuçlar da daha detaylı bir şekilde açıklanacaktır.

Bu blog sitesi projesi, kullanıcıların interaktif bir ortamda blogları keşfetmelerine olanak sağlayarak, bilgi paylaşımı ve etkileşimi teşvik etmeyi amaçlamaktadır. Kullanıcılar, ilgi duydukları farklı kategorilerdeki blogları kolaylıkla bulabilecek ve okuyabilecekler. Ayrıca blog yazarlarıyla etkileşime geçebilecek, yorumlar yapabilecek, beğenilerini ve beğenmeme ifadelerini paylaşabilecekler.

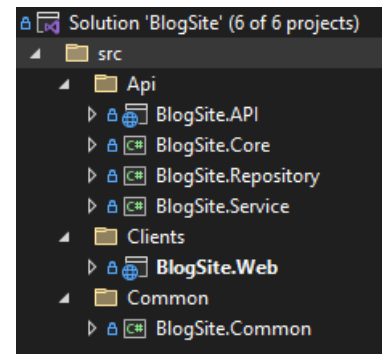
Bu projenin hedefi, kullanıcıların kişisel ilgi alanlarına uygun içerikleri keşfetmelerini sağlamak ve aynı zamanda blog yazarlarıyla etkileşime geçerek bir topluluk oluşturmaktır. Kullanıcılar, kendi favori bloglarını kaydedebilecek, istedikleri içerikleri diğer kullanıcılarla paylaşabilecek ve blog yazarlarıyla doğrudan iletişim kurabilecekler. Bu şekilde, farklı düşüncelerin ve bakış açılarının paylaşılmasıyla zenginleşen bir blog deneyimi sunmayı hedefliyoruz.

Bu projenin ayrıca blog yazarlarına da büyük bir fırsat sunması hedeflenmektedir. Yazarlar, kendi bloglarını platformda paylaşarak geniş bir okuyucu kitlesine erişebilecek, yazıları hakkında geri bildirim alabilecek ve okuyucularla etkileşime geçerek tartışmalara katılma imkanı bulabilecekler. Aynı zamanda, yazarlar kendilerini tanıtabilecekleri bir profil oluşturabilecek ve takipçileriyle doğrudan iletişim kurabilecekler. Bu proje, blog yazarlarının kendilerini daha geniş bir kitleye duyurmasını ve takipçi tabanını artırmasını amaçlamaktadır, böylece yazarlar daha fazla kişiye ulaşarak etkilerini ve etkileşimlerini artıracaklardır.

II. YAZILIM MIMARISI

Bu projede, n-layer bir mimari yaklaşımı benimsenmiştir. Çekirdek yapıların yer aldığı Core katmanı, projenin temel yapılarını içermektedir. Veritabanı işlemleri ve veri tabanı modellerinin konfigürasyonu Repository katmanında gerçekleştirilirken, Service katmanı ise iş mantığı işlemlerini yönetmektedir. API katmanı, dış dünyayla iletişimi sağlayarak kullanıcılara web API olarak hizmet verirken, .NET Core MVC web katmanı kullanıcıların arayüzde etkileşimde bulunabilecekleri kısmı sunmaktadır.

Bu n-layer mimari yapısı, projenin modüler olmasını sağlayarak bakım süreçlerini kolaylaştırmaktadır. Her bir katmanın belirli bir sorumluluğu ve işlevi vardır, bu da kodun daha okunabilir, sürdürülebilir ve genişletilebilir olmasını sağlamaktadır. Bu mimari yapının kullanılması, farklı projelerde de tercih edilebilir ve projelerin geliştirilmesinde hız, verimlilik ve düzen sağlar.

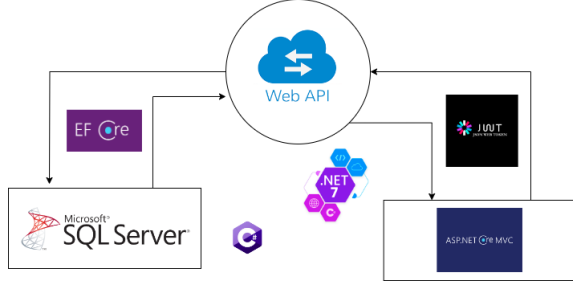


Şekil 1. Projenin solution yapısı

Her bir katman, belirli bir sorumluluğa odaklanarak projenin daha iyi organize edilmesini sağlar. Core katmanı, projenin çekirdek yapılarını ve paylaşılan bileşenleri içererek genel bir yapı oluşturur. Bu sayede, projede kullanılan ortak bileşenlerin tekrar kullanılması ve birbiriyle entegrasyonu kolaylaşır.

Repository ve Service katmanları, veri tabanı işlemlerini ve iş mantığı işlemlerini yöneterek projenin veri tabanı

bağlantısını ve iş süreçlerini sağlamlaştırır. Repository katmanı, generic repository tasarım deseni kullanarak veritabanı işlemlerini soyutlar ve herhangi bir nesne için CRUD işlemlerini gerçekleştirmeyi sağlar. Service katmanı ise, bu veri tabanı işlemlerini kullanarak iş mantığı işlemlerini gerçekleştirir ve kullanıcıların isteklerine yanıt verir.



Şekil 2. Yazılım mimarisi

Bu projede, API katmanı hem JWT token dağıtımını hem de doğrulama işlemlerini gerçekleştirmektedir. API katmanında, kullanıcıların kimlik doğrulamasını sağlamak ve yetkilendirme işlemlerini gerçekleştirmek için JWT (JSON Web Token) kullanılmaktadır. JWT token, kullanıcı kimlik bilgilerini içeren bir JSON veri yapısıdır ve kullanıcının doğrulama durumunu ve yetkilendirmesini taşır.

API katmanı, kullanıcının giriş yapması durumunda, kullanıcı kimlik bilgilerini doğrular ve geçerli bir JWT token oluşturur. Bu token, kullanıcıya döndürülerek tarayıcı veya diğer istemcilerle paylaşılır. Kullanıcı, bu JWT token'ı alarak MVC tarafına geçer.

MVC tarafında, gelen JWT token ile authentication (kimlik doğrulama) işlemi gerçekleştirilir. MVC projesinde, gelen JWT token doğrulanır ve token içerisindeki kullanıcı kimlik bilgileri kullanılarak kullanıcının oturumu oluşturulur. Böylece, kullanıcı API katmanındaki yetkilendirme sürecini geçmiş olur ve uygulama içinde yetkilendirilmiş işlemleri gerçekleştirebilir.

Bu yapı sayesinde, API katmanı JWT token dağıtarak kullanıcıların kimlik doğrulama sürecini yönetirken, MVC tarafı gelen JWT token'ı doğrulayarak kullanıcının uygulama içinde yetkilendirilmiş işlemleri gerçekleştirmesine olanak sağlar. Bu, güvenli ve yetkilendirilmiş bir kullanıcı deneyimi sunmak için etkili bir mimari yaklaşım sağlar.

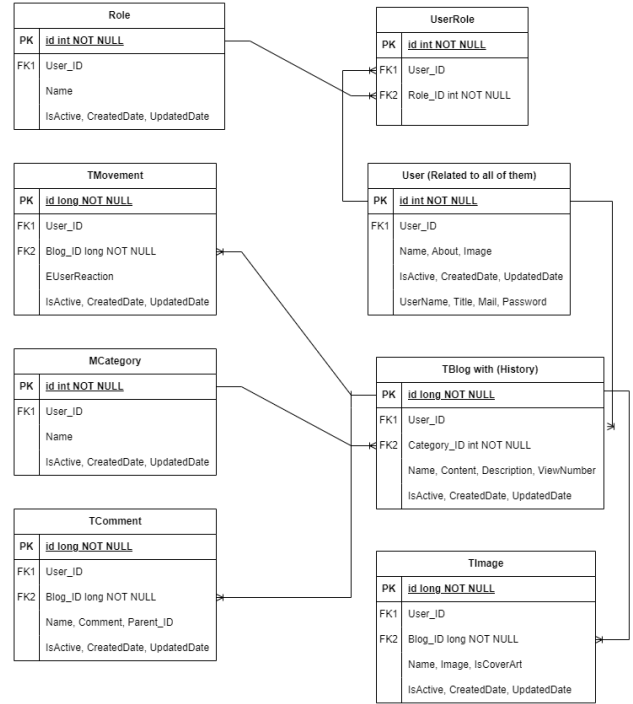
MVC tarafında, kullanıcılar ve adminler için farklı template'lerin kullanıldığı bir yapı oluşturulmuştur. Bu, kullanıcı ve admin kullanıcı arayüzlerinin farklılıklarını ve ihtiyaçlarını karşılamak amacıyla gerçekleştirilmiştir. Kullanıcılar için bir template, adminler için ise farklı bir template kullanılmaktadır. Bu sayede, kullanıcılara özgü işlevler ve kullanıcı deneyimi sağlanırken, adminlerin yetkilendirme ve yönetim işlemleri için ayrı bir arayüz sunulmaktadır.

Projede JWT token'larının saklanması için Cookie kullanılmıştır. JWT token, kullanıcı oturumu boyunca güvenli bir şekilde saklanabilmesi için tarayıcıdaki Cookie mekanizmasında tutulmaktadır. Bu sayede, her istekte token almak için ayrı bir istek yapmaktan kaçınılarak, istekler daha hızlı ve verimli bir şekilde gerçekleştirilebilir.

Her istekte, JWT token'ını almak ve HTTP isteğine eklemek için bir Middleware kullanılmıştır. Bu Middleware, her gelen isteği kontrol eder ve kullanıcının tarayıcısındaki Cookie'den JWT token'ını alarak, ilgili HTTP isteğine ekler. Böylece, her istekte kullanıcının oturum bilgileri doğru şekilde API'ye iletilir ve yetkilendirme süreci gerçekleştirilir. Bu yapı, güvenliği sağlamak ve kullanıcı deneyimini iyileştirmek için etkili bir şekilde kullanılmaktadır.

API ve .NET Core MVC web katmanları, kullanıcılarla etkileşim kurulan ve sunulan hizmetlerin görüntülediği katmanlardır. API katmanı, web API olarak tasarlanmıştır ve kullanıcıların taleplerini karşılar. .NET Core MVC web katmanı ise, kullanıcıların arayüzde etkileşimde bulunabilecekleri ve sunulan içerikleri görüntüleyebilecekleri kısmı sağlar. Bu katmanlar, kullanıcı deneyimini iyileştirmek ve etkileşimi kolaylaştırmak için önemli rol oynamaktadır.

Her bir katman, bağımsız olarak test edilebilir ve hataların izlenmesi ve giderilmesi daha etkili hale gelir. Core katmanı, projenin çekirdek yapılarını içerdiği için temel işlevlerin doğru çalıştığından emin olmak için test edilebilir. Repository katmanı, veri tabanı işlemlerinin doğru şekilde gerçekleştirildiğini ve modellerin uygun şekilde konfigüre edildiğini doğrulamak için test edilebilir. Service katmanı, iş mantığı işlemlerinin beklenen sonuçları ürettiğini test etmek için uygundur.



Şekil 3. Veri tabanı diyagramı

API ve .NET Core MVC web katmanları da test edilebilirlik açısından önemli rol oynamaktadır. API katmanı, gelen talepleri doğru şekilde işlediğini ve beklenen yanıtları döndürdüğünü doğrulamak için test edilebilir. .NET Core MVC web katmanı, kullanıcı arayüzünün doğru çalıştığını ve sunulan hizmetlerin etkileşimli bir şekilde kullanıcılar tarafından kullanılabilirliğini test etmek için uygun bir katmandır. Her bir katmanın test edilebilirliği, yazılımın kalitesini artırır ve sorunların erken tespit edilmesini sağlar.

III. KULLANILAN TEKNOLOJILER

Bu projede, web uygulamasının geliştirilmesi için çeşitli teknolojiler kullanılmıştır. Front-end tarafında HTML, CSS ve JavaScript ile birlikte jQuery kütüphanesi kullanılmıştır. Bu teknolojiler, kullanıcı arayüzünün tasarlanması ve etkileşimli öğelerin oluşturulması için kullanılmıştır.

Back-end tarafında ise .NET kullanılmıştır. Proje, .NET Core MVC projesi olarak geliştirilmiştir. Bu sayede, kullanıcıların isteklerini alarak işlemleri gerçekleştiren bir sunucu tarafı oluşturulmuştur. .NET Core, platform bağımsızlığı ve yüksek performansı ile öne çıkan bir framework'tür.

Veritabanı olarak Microsoft SQL Server tercih edilmiştir. Veritabanı işlemleri için Entity Framework Core kullanılmıştır. Bu ORM (Object-Relational Mapping) teknolojisi, veritabanı işlemlerini kolaylaştırırken, veri tabanı nesneleriyle program arasındaki ilişkiyi yönetmektedir.

Projenin mimarisini desteklemek için n-layer mimari kullanılmıştır. Core, repository, service, API ve .NET Core MVC web katmanları oluşturulmuştur. Bu katmanlar, kodun daha modüler ve bakımı kolay hale getirilmesini sağlar. Ayrıca, projenin güvenliği için JWT (JSON Web Token) kullanılmıştır. JWT token'larının saklanması için Cookie mekanizması kullanılmış ve her istekte Middleware ile token doğrulama işlemi gerçekleştirilmiştir.

Kullanılan teknolojiler, performans, güvenlik ve kullanıcı deneyimi gibi önemli faktörleri göz önünde bulundurarak projenin başarıyla tamamlanmasını sağlamıştır.

IV. KOLAY KULLANIM



Şekil 4. Giriş ve kayıt ekranı

Şekil 4 de gösterilen ekran üzerinde kullanıcı hem giriş yapabilir yada yeni kayıt olabilir.



Şekil 5. Anasayfa



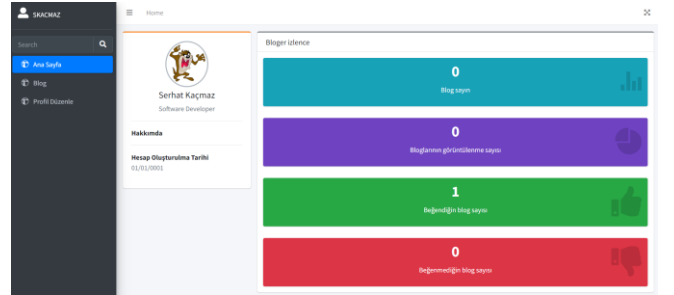
Şekil 6. Kullanıcı reaksiyonu

Şekil 6 da kullanıcı okuduğu blog yazısını beğendiğinde, beğenmediğinde veya favori olarak eklemek isteğinde burdaki ekrandan tercihini işaretler.



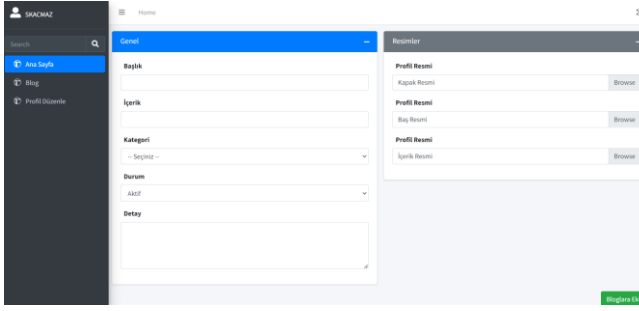
Şekil 7. Blog içerik

Şekil 7 de blog yazısının detayı ve okuyucu tarafından yorum bırakılmak istenildiği zaman yorum bırakabileceği ekranı gösterir.



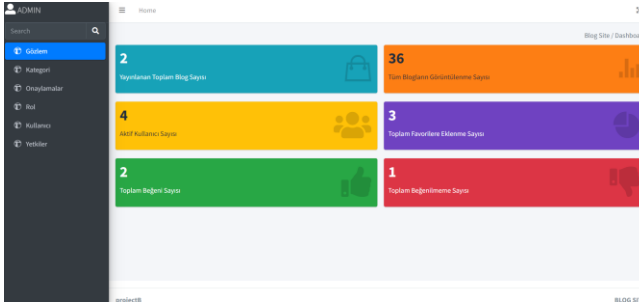
Şekil 8. Kullanıcı dashboard ekranı

Şekil 8 deki ekranda kullanıcıya sistem üzerinde yapmış olduğu bazı işlemlerin istatistikleri verilir. Bu Dashboard özelinde blog ekleyebilir, profilini güncelleyebilir ve işaretlediği bloglara gidebilir gibi seçenekleri kullanıcıya sunar.



Şekil 9. Blog oluşturma ekranı

Şekil 9 daki görsel, kullanıcılar için blog yazıp sistemde yayınlamaları için kullanılması gerekli olan ekranı gösterir.



Şekil 10. Yönetici dashboard ekranı

Şekil 10 daki ekranda yöneticiye sistem üzerindeki belli başlı bazı işlemlerin istatistiklerini sunar. Toplam kullanıcı sayısı, toplam beğeni, toplam blog sayısı gibi sayısal değerlere erişebilir.

V. SONUÇ

Bu proje, kullanıcıların birden fazla kategoride blogları okuyabildiği, yorum yapabildiği, beğeni ve beğenmeme ifadesi verebildiği, favorilere ekleyebildiği ve paylaşabildiği bir blog sitesi platformu geliştirmeyi amaçlamaktadır. Geliştirilen platform, kullanıcıların ilgi duydukları konularda blogları okuyabilecekleri, yazarlarla etkileşime geçebilecekleri ve fikirlerini paylaşabilecekleri bir ortam sunmaktadır.

Proje, n-layer mimari yapısıyla tasarlanmış ve güncel teknolojiler kullanılarak geliştirilmiştir. Bu mimari, projenin modülerliğini ve bakımını kolaylaştırmıştır. Kullanılan teknolojiler, performans, güvenlik ve kullanıcı deneyimi gibi önemli faktörleri gözetenek projenin başarılı bir şekilde tamamlanmasını sağlamıştır.

Projenin hedefleri doğrultusunda, kullanıcılar ve adminler için farklı template'ler tasarlanmış ve kullanılmıştır. Bu sayede, kullanıcıların ve adminlerin ihtiyaçlarına uygun bir arayüz sunulmuştur. JWT token'ları, Cookie mekanizmasında saklanarak güvenli bir şekilde kullanıcı oturumları yönetilmiştir.

Proje tamamlandığında, belirlenen hedeflerin başarıyla gerçekleştirildiği görülmüştür. Kullanıcılar, istedikleri kategorideki blogları okuyabilir, yorum yapabilir ve beğeni ifadesi verebilirken, adminler ise yönetici işlemlerini gerçekleştirebilmiştir.

Gelecekte, projenin daha da geliştirilerek yeni özelliklerin eklenmesi önerilmektedir. Örneğin, arama ve filtreleme özelliği gibi kullanıcı deneyimini artıracak özelliklerin entegre edilmesi düşünülebilir. Ayrıca, daha kapsamlı bir analitik sistem oluşturularak kullanıcıların tercihlerini daha iyi anlamak ve içeriklerini kişiselleştirmek mümkün olabilir.

Sonuç olarak, bu proje, kullanıcılar için ilgi çekici bir blog sitesi platformu oluşturma hedefini başarıyla gerçekleştirmiştir. Kullanılan teknolojiler, mimari yapısı ve özellikleriyle projenin güvenli, modüler ve kullanıcı dostu bir yapıya sahip olduğunu göstermektedir.

VI. KAYNAKÇA

- <https://learn.microsoft.com/en-us/dotnet/>
- <https://jwt.io/>
- <http://cagataykiziltan.net/automapper-kullanimi-ve-dto/>
- <https://www.gencayyildiz.com/blog/net-core-fluentvalidation-ile-validasyon-kontrolu/>
- <https://www.youtube.com/@TechBuddyTR>