

**ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM4531 - AĞ TABANLI TEKNOLOJİLER VE
UYGULAMALARI PROJE RAPORU**

KAMU KÖPRÜSÜ
Şikayet ve Öneri Portalı

<https://youtu.be/Fc21jjjVNZQ>

Aybaran Yurtseven - 22290067

10.01.2026

ÖZET

Bu rapor; Ağ Tabanlı Teknolojiler ve Uygulamaları dersi kapsamında geliştirilen, vatandaşların kamu kurumlarına şikayet ve önerilerini iletebileceği, medya dosyaları ekleyebileceği ve kurumların çözüm süreçlerini takip edebileceği "Kamu Köprüsü" adlı web tabanlı yazılım sisteminin tasarım, geliştirme ve uygulama süreçlerini kapsamlı bir şekilde açıklamak amacıyla hazırlanmıştır. Proje; vatandaş-kurum iletişimini dijitalleştirme, basıyuru süreçlerini şeffaflaştırma ve kullanıcı katılımını oyunlaştırma (gamification) mekanizmalarıyla teşvik etmeyi hedeflemektedir.

Yazılım geliştirme süreci, modern yazılım yaşam döngüsü prensiplerine uygun olarak GitHub üzerinde versiyon kontrolü ile yönetilmiştir. Sistemin teknik altyapısında Katmanlı Mimari prensibiyle .NET 10.0 çerçevesi kullanılarak MVC (Model-View-Controller) tasarım deseni uygulanmış; veri güvenliği ve bütünlüğü SQL Server veri tabanı ve Entity Framework Core 10.0 ORM aracı ile sağlanmıştır. Kullanıcı arayüzü Razor Views teknolojisiyle dinamik ve kullanıcı dostu bir yapıda tasarlanmış, güvenlik ASP.NET Core Identity altyapısı ve rol tabanlı yetkilendirme mekanizmalarıyla üst düzeyde sağlanmıştır.

Sistem; beş farklı kullanıcı rolü (Vatandaş, Kurum Temsilcisi, STÖ Koordinatörü, Moderatör, Admin), kapsamlı bir rozet ve seviye sistemi, çoklu medya desteği (fotoğraf, video, ses) ve RESTful API yapısı ile güçlü bir altyapı sunmaktadır. Ayrıca Swagger/OpenAPI entegrasyonu ile API dokümantasyonu sağlanmış, sistemin ölçeklenebilirliği ve bakım kolaylığı garanti altına alınmıştır.

Bu çalışmada, e-devlet uygulamalarına modern web teknolojilerinin entegrasyonu alanında bir örnek teşkil etmekte olup geliştirilen sistemin sürdürülebilir, ölçeklenebilir ve yüksek performanslı bir yazılım altyapısı sunduğu gösterilmektedir.

İçindekiler

1 GİRİŞ	4
1.1 Kamu Köprüsü Nedir	4
1.2 Amaç ve Kapsam	4
1.3 Çalışmanın Önemi	5
1.4 Raporun Yapısı	5
2 PROJE ORTAMI VE KULLANILAN TEKNOLOJİLER	7
2.1 Geliştirme Ortamı	7
2.2 Projenin Genel Teknik Yapısı	7
2.3 Kullanılan Yazılım Teknolojileri	8
2.4 Sistem Mimarisi	8
3 YAZILIM GERÇEKLEME SÜRECİ	10
3.1 Geliştirilen Modüller.....	10
3.1.1 Kimlik Doğrulama ve Yetkilendirme Modülü	10
3.1.2 Vatandaş Modülü	10
3.1.3 Kurum Temsilcisi Modülü	11
3.1.4 Moderatör Modülü	11
3.1.5 Yönetici (Admin) Modülü	11
3.1.6 Oyunlaştırma (Gamification) Modülü	12
3.2 Model-View-Controller Tasarımı.....	12
3.2.1 Controllers (Denetleyiciler)	12
3.2.2 Views (Görünümler)	13
3.3 Veri Tabanı Tasarımı	13
3.3.1 Varlık Modelleri (Entity Models)	13
3.3.2 Veritabanı İlişkileri.....	14
3.3.3 DbContext Yapılandırması	14
3.4 Kullanıcı Kimlik Doğrulama	14
3.4.1 Şifre Politikası	15
3.4.2 Hesap Kilitleme	15
3.4.3 Rol Tabanlı Yetkilendirme	15

3.4.4 Oturum Yönetimi	15
4 DEĞERLENDİRME	16
4.1 Kullanılabilirlik Değerlendirmesi	16
4.2 Teknik Değerlendirme.....	16
5 SONUÇ	18
6 KAYNAKLAR	19
6.1 Uluslararası Standartlar	19
6.2 Teknik Dokümantasyon ve Kılavuzlar	19

1 GİRİŞ

Günümüzde dijital dönüşüm sürecinde vatandaş-devlet etkileşiminin modernizasyonu kritik bir öneme sahiptir. Geleneksel şikayet ve öneri mekanizmalarının yetersiz kaldığı, başvuru süreçlerinin takip edilemediği ve şeffaflığın sağlanamadığı ortamlarda vatandaş memnuniyetsizliği artmaktadır. Bu bağlamda geliştirilen "Kamu Köprüsü" projesi, bu problemleri çözmek ve vatandaş-kurum iletişimini dijital bir platforma taşımak amacıyla tasarlanmış, bütünlüklük bir web çözümüdür.

1.1 Kamu Köprüsü Nedir

Kamu Köprüsü, vatandaşların kamu kurumlarına şikayet ve önerilerini dijital ortamda iletebileceği, fotoğraf, video ve ses dosyaları ekleyebileceği, başvuru durumlarını gerçek zamanlı takip edebileceği kapsamlı bir e-devlet portalıdır. Sistem, geleneksel dilekçe süreçlerinin ötesine geçerek;

- Vatandaşlara kolay ve hızlı başvuru imkanı sunar
- Kurumlara etkin başvuru yönetim araçları sağlar
- Moderatör denetimi ile içerik kalitesini güvence altına alır
- Oyunlaştırma (gamification) mekanizmalarıyla kullanıcı katılımını teşvik eder

Platform; konu tabanlı şikayet/öneri sistemi, çoklu medya desteği, rozet ve seviye sistemi, moderasyon altyapısı ve kapsamlı yönetici paneli olmak üzere beş temel fonksiyon üzerine inşa edilmiştir.

1.2 Amaç ve Kapsam

Bu projenin temel amacı; vatandaşların kamu kurumlarıyla iletişimini kolaylaştıran, başvuru süreçlerini şeffaflaştıran ve kurumların vatandaş geri bildirimlerini etkin şekilde yönetebilmesini sağlayan teknolojik bir ekosistem oluşturmaktır. Proje kapsamı şu ana başlıkları içermektedir:

- **Çok Rollü Kullanıcı Yönetimi:** Vatandaş, Kurum Temsilcisi, STÖ Koordinatörü, Moderatör ve Admin olmak üzere beş farklı kullanıcı rolü için özelleştirilmiş arayüzler ve yetkiler tanımlanmıştır.

- **Kapsamlı Şikayet/Öneri Sistemi:** Kullanıcıların detaylı başvuru ve açıklama ile başvuru oluşturabileceği, kategori seçebileceği, kurum belirleyebileceği ve çoklu medya dosyası ekleyebileceği dinamik bir içerik yönetim sistemi geliştirilmiştir.
- **Oyunlaştırma Mekanizması:** Kullanıcı katılımını artırmak amacıyla puan, seviye (Bronze, Silver, Gold, Platinum, Diamond) ve rozet sistemi entegre edilmiştir.
- **Güvenli Veri Yönetimi:** Kullanıcı verilerinin ve platform içeriklerinin korunması amacıyla ASP.NET Core Identity tabanlı kimlik doğrulama, rol tabanlı yetkilendirme ve hesap kilitleme mekanizmaları kullanılarak güvenlik optimize edilmiştir.
- **Moderasyon Altyapısı:** İçerik kalitesini korumak için dört aşamalı uyarı sistemi ve moderatör inceleme süreçleri tasarlanmıştır.

1.3 Çalışmanın Önemi

Yazılım dünyasında teknolojiler hızla evrilirken geliştirilen projelerin güncel standartlara uyum sağlaması gerekmektedir. Kamu Köprüsü, kullanılan teknoloji yığını (Tech Stack) bakımından endüstrinin en güncel sürümleri olan .NET 10.0 ve Entity Framework Core 10.0 altyapılarını barındırmasıyla teknik bir özgünlük taşımaktadır.

Sosyal açıdan ise vatandaş, -kurum iletişimini merkezi bir platformda toplayarak şeffaflık ve hesap verebilirlik sağlaması, oyunlaştırma mekanizmalarıyla kullanıcı katılımını teşvik etmesi projenin sektörel ve akademik önemini artırmaktadır. Ayrıca RESTful API yapısı ve Swagger dokümantasyonu ile mobil uygulama veya üçüncü parti entegrasyonlara açık bir altyapı sunulmaktadır.

1.4 Raporun Yapısı

Kamu Köprüsü projesinin tasarım, geliştirme ve uygulama süreçlerini detaylandıran bu rapor beş ana bölümden oluşmaktadır:

Raporun birinci bölümünde; Kamu Köprüsü platformunun hem teknik hem de sosyal yönü ele alınmış, ve çalışmanın amacı, kapsamı, önemi açıklanarak proje için kuramsal bir temel oluşturulmuştur.

Raporun ikinci bölümünde; projenin geliştirildiği donanım ve yazılım ortamı, tercih edilen programlama dilleri, kütüphaneler ve sistemin genel mimari yapısı açıklanmıştır.

Raporun üçüncü bölümünde; yazılımın gerçekleştirmesi süreci, MVC tasarımı, veritabanı tasarımı ve geliştirilen modüller teknik detaylarla incelenmiştir.

Raporun dördüncü bölümünde; geliştirilen sistemin kullanılabilirlik değerlendirilmesi yapılmıştır, projenin güçlü yönleri ve kısıtları ele alınmıştır.

Raporun beşinci bölümünde; proje sürecinden elde edilen kazanımlar ve genel sonuçlar özetlenmiştir.

2 PROJE ORTAMI VE KULLANILAN TEKNOLOJİLER

Bu bölümde; Kamu Köprüsü projesinin tasarım ve geliştirme süreçlerinde kullanılan donanım ve yazılım ortamları, kaynak kod yönetim stratejileri, test süreçleri, tercih edilen programlama dilleri ve sistemin genel mimari yapısı detaylı olarak ele alınmıştır.

2.1 Geliştirme Ortamı

Projenin kodlama, hata ayıklama ve sürüm kontrolü süreçlerinde verimliliği artırmak ve endüstri standartlarını yakalamak amacıyla profesyonel geliştirme araçları tercih edilmiştir. Geliştirme sürecinde .NET ekosistemiyle tam uyumlu çalışan Visual Studio 2022 ve performansı ile geniş eklenti desteği nedeniyle Visual Studio Code kullanılmıştır.

Projenin geliştirme sürecinde kod yönetimi ve sürüm takibinin etkin bir şekilde sağlanabilmesi amacıyla GitHub platformu kullanılmıştır. Geliştirme süreci boyunca yapılan çalışmalar, düzenli ve açıklayıcı işlem kayıtları Git sayesinde commit’lenerek sisteme işlenmiştir; bu yaklaşım yazılım bileşenlerinin izlenebilir, geri alınabilir ve düzenli bir yapıda yönetilmesine olanak tanımıştır.

Yazılım bileşenlerinin test edilmesi ve doğrulanması işlemleri yerel geliştirme ortamlarında (Localhost) gerçekleştirilmiştir. Her bir güncelleme ve kod entegrasyonunun ardından sistemin temel işlevleri kontrol edilerek geliştirilen modüllerin birbiriyle uyumlu ve hatasız çalışması garanti altına alınmıştır. API uç noktalarının doğrulanması için Swagger UI kullanılmıştır.

2.2 Projenin Genel Teknik Yapısı

Geliştirilen sistem modern web yazılım standartlarına uygun olarak MVC (Model-View-Controller) mimarisi üzerine inşa edilmiştir. Sistem; Sunum Katmanı (Razor Views, Controllers), İş Mantığı Katmanı (Services, Validators) ve Veri Erişim Katmanı (Entity Framework Core, DbContext) olmak üzere üç temel katmandan oluşmaktadır.

İstemci ve sunucu arasındaki veri iletişimi, platform bağımsızlığını ve farklı sistemlerle entegrasyon yeteneğini sağlamak adına RESTful API mimarisi kullanılarak gerçekleştirilmiştir. API dokümantasyonu Swagger/OpenAPI ile otomatik olarak oluşturulmaktadır.

2.3 Kullanılan Yazılım Teknolojileri

Projenin teknoloji yığı (Tech Stack), performans, güvenlik, sürdürülebilirlik ve veri tutarlılığı gereksinimleri gözetilerek üretildikleri teknolojilerin en güncel kararlı sürümlerinden oluşturulmuştur:

- **Framework:** Microsoft tarafından geliştirilen ASP.NET Core MVC, .NET 10.0 platformu üzerinde sunucu tarafı iş mantığının yürütülmesi için kullanılmıştır.
- **ORM (Object-Relational Mapping):** Veri tabanı ile nesne yönelimli programlama arasındaki köprüyü kurmak amacıyla Entity Framework Core 10.0 tercih edilmiştir. Code-First yaklaşımı ile veritabanı şeması C# sınıflarından türetilmiştir.
- **Kimlik Doğrulama:** ASP.NET Core Identity 10.0 kullanılarak kullanıcı yönetimi, rol tabanlı yetkilendirme ve güvenlik özellikleri sağlanmıştır.
- **Veri Tabanı Sistemi:** Verilerin güvenli, tutarlı ve ilişkisel bir yapıda saklanması amacıyla SQL Server LocalDB kullanılmıştır.
- **API Dokümantasyonu:** Swagger/OpenAPI 7.2.0 ile otomatik API dokümantasyonu ve test arayüzü sağlanmıştır.
- **Ön Yüz Teknolojileri:** Razor Views, Bootstrap CSS framework ve JavaScript kullanılarak dinamik ve responsive bir kullanıcı arayüzü oluşturulmuştur.

Tablo 1: Teknoloji Yığını Özeti

Kategori	Teknoloji	Versiyon
Framework	ASP.NET Core MVC	.NET 10.0
ORM	Entity Framework Core	10.0.0
Veritabanı	SQL Server (LocalDB)	-
Kimlik Doğrulama	ASP.NET Core Identity	10.0.0
API Dokümantasyonu	Swagger/OpenAPI	7.2.0

2.4 Sistem Mimarisi

Yazılımın bakım kolaylığını, test edilebilirliğini ve genişletilebilirliğini sağlamak amacıyla proje genelinde Katmanlı Mimari (Layered Architecture) deseni uygulanmıştır. Bu mimari yaklaşım

sayesinde sistemin farklı sorumlulukları birbirinden izole edilerek modüler bir yapı elde edilmiştir:

- **Sunum Katmanı (Presentation Layer):** İstemciden gelen HTTP isteklerinin karşılanması ve kullanıcı arayüzünün sunulduğu katmandır. Razor Views ve MVC/API Controllers bu katmanda yer alır.
- **İş Mantığı Katmanı (Business Layer):** Uygulamanın temel iş kurallarının ve mantıksal süreçlerin yürütüldüğü katmandır. GamificationService (puan ve rozet yönetimi) ve AuditService (denetim kaydı) gibi servisler bu katmanda yer alır.
- **Veri Erişim Katmanı (Data Layer):** Veri tabanı CRUD (Oluşturma, Okuma, Güncelleme, Silme) işlemlerinin gerçekleştirildiği katmandır. ApplicationDbContext ve Entity Framework Core bu katmanda çalışır.
- **Veri Tabanı Mimarisi:** Proje, veriler arasındaki bütünlüğü korumak adına ilişkisel Veri Tabanı (Relational Database) modelini benimsemiştir. Bir-Çok (One-to-Many) ve Bir-Bir (One-to-One) ilişkiler kurularak veri tutarlılığı sağlanmıştır.

3 YAZILIM GERÇEKLEME SÜRECİ

Kamu Köprüsü projesinin gerçekleştirilme sürecinde modern yazılım geliştirme yaşam döngüsü prensiplerine sadık kalınmış,; sürdürülebilirlik, ölçeklenebilirlik ve güvenlik unsurları ön planda tutulmuştur.

Geliştirme sürecinde veritabanı seçiminin oluşturulmasıyla başlayan süreç, iş mantığı kurallarının servis katmanlarında tanımlanması ve son kullanıcı etkileşimini sağlayan arayüz bileşenlerinin entegrasyonu ile tamamlanmıştır. Kodlama aşamasında SOLID prensiplerine ve "Temiz Kod" (Clean Code) standartlarına özen gösterilerek modüler ve geliştirilmeye açık bir yapı hedeflenmiştir.

3.1 Geliştirilen Modüller

Kamu Köprüsü sistemi, kullanıcıların farklı ihtiyaçlarına yanıt verebilmek ve yönetsel süreçleri kolaylaştırmak amacıyla belirli sorumluluklara sahip, birbirinden bağımsız ancak entegre çalışabilen modüller halinde geliştirilmiştir.

3.1.1 Kimlik Doğrulama ve Yetkilendirme Modülü

Sistemin güvenliğini sağlamak ve kişiselleştirilmiş bir kullanıcı deneyimi sunmak amacıyla AccountController üzerinden kimlik doğrulama modülü tasarlanmıştır. Bu modülde:

- Kullanıcı kayıt (Register) ve giriş (Login) süreçleri yönetilmektedir
- Şifreler BCrypt benzeri güvenli algoritmalar ile hashlenmektedir
- Beş farklı rol için yetkilendirme politikaları tanımlanmıştır
- Hesap kilitleme mekanizması (5 başarısız deneme - 10 dakika kilitleme) aktiftir
- Türkçe hata mesajları TurkishIdentityErrorDescriber ile sağlanmaktadır

3.1.2 Vatandaş Modülü

CitizenController üzerinden yönetilen bu modül, vatandaşların temel işlevlerini kapsar:

- **Dashboard:** Kişisel istatistikler, başvuru özeti ve seviye bilgisi

- **CreateComplaint:** Yeni şikayet/öneri oluşturma (başlık, açıklama, kategori, kurum seçimi, medya ekleme)
- **MyComplaints:** Kullanıcının kendi başvurularını listeleme ve filtreleme
- **ComplaintDetail:** Başvuru detaylarını ve durum güncellemelerini görüntüleme
- **Achievements:** Kazanılan rozetler ve seviye ilerlemesi

3.1.3 Kurum Temsilcisi Modülü

InstitutionController üzerinden yönetilen bu modül, kurum temsilcilerinin başvuru yönetim süreçlerini kapsar:

- **Dashboard:** Kurum istatistikleri ve bekleyen başvuru sayısı
- **IncomingComplaints:** Kuruma gelen başvuruları listeleme ve filtreleme
- **ManageComplaint:** Başvuru durumu güncelleme ve geri bildirim yazma
- **ResolvedComplaints:** Çözülen başvuruların arşivi

3.1.4 Moderatör Modülü

ModeratorController üzerinden yönetilen bu modül, içerik denetimi ve kullanıcı yönetimi süreçlerini kapsar:

- **Dashboard:** Moderasyon istatistikleri ve bekleyen içerikler
- **ReviewContent:** Onay bekleyen başvuruları inceleme
- **IssueWarning:** Kullanıcılara uyarı verme (4 aşamalı sistem)
- **BanUser:** Kural ihlali durumunda kullanıcı yasaklama

3.1.5 Yönetici (Admin) Modülü

AdminController üzerinden yönetilen bu modül, sistem genelini kapsayan yönetimsel işlevleri içerir:

- **Dashboard:** Sistem geneli istatistikler ve grafikler

- **Users:** Kullanıcı yönetimi (listeleme, düzenleme, silme)
- **Complaints:** Tüm bas_vuruları yönetme
- **InstitutionApprovals:** Kurum temsilcisi onay/red iş_lemeleri
- **Reports:** Raporlar ve analitik veriler

3.1.6 Oyunlaştırma (Gamification) Modülü

GamificationService üzerinden yönetilen bu modül, kullanıcı katılımını teş_vik eden mekanizmaları kapsar:

- **Puan Sistemi:** İlk bas_vuru (+10), onaylanan bas_vuru (+15), çözülen bas_vuru (+25), medya ekleme (+5)
- **Seviye Sistemi:** Bronze (0-99), Silver (100-249), Gold (250-499), Platinum (500-999), Diamond (1000+)
- **Rozet Sistemi:** 8 farklı rozet (Söz Sahibi, Fark Yaratıcı, Sürekli Çabalar vb.)

3.2 Model-View-Controller Tasarımı

Kamu Köprüsü projesinin mimarisi, MVC (Model-View-Controller) tasarım deseni üzerine inşa edilmiştir. Bu yapı sayesinde iş_mantığı, veri yönetimi ve sunum katmanları birbirinden ayrılmıştır.

3.2.1 Controllers (Denetleyiciler)

Projede altı ana MVC Controller ve üç API Controller geliştirilmiştir:

Tablo 2: MVC Controller Yapısı

Controller	Rol	Ana Fonksiyonlar
HomeController	Genel	Anasayfa, İletişim, Hakkında
AccountController	Kimlik	Login, Register, Logout
CitizenController	Vatandaş	Dashboard, Şikayet Oluşturma
InstitutionController	Kurum	Bas_vuru Yönetimi
ModeratorController	Moderatör	İçerik Denetimi
AdminController	Admin	Sistem Yönetimi

Tablo 3: API Controller Yapısı

Controller	Endpoint'ler
ComplaintsApiController	GET, POST, PUT, DELETE /api/complaints
InstitutionsApiController	GET /api/institutions
UsersApiController	GET, PUT /api/users

3.2.2 Views (Görünümler)

Kullanıcı arayüzü Razor Views teknolojisi ile geliştirilmiştir. Her rol için ayrı view dizinleri oluşturulmuştur:

- **Account/:** Login.cshtml, Register.cshtml, AccessDenied.cshtml
- **Citizen/:** Dashboard.cshtml, CreateComplaint.cshtml, MyComplaints.cshtml, ComplaintDetail.cshtml, Achievements.cshtml, Profile.cshtml, EditProfile.cshtml
- **Institution/:** Dashboard.cshtml, IncomingComplaints.cshtml, ManageComplaint.cshtml, ResolvedComplaints.cshtml
- **Moderator/:** Dashboard.cshtml, ReviewContent.cshtml, IssueWarning.cshtml, BannedUsers.cshtml
- **Admin/:** Dashboard.cshtml, Users.cshtml, Complaints.cshtml, InstitutionApprovals.cshtml, EditUser.cshtml, AllComplaints.cshtml, Reports.cshtml
- **Shared/:** _Layout.cshtml, _ValidationScriptsPartial.cshtml, Error.cshtml, _StatusBadge.cshtml

3.3 Veri Tabanı Tasarımı

Projenin veri saklama altyapısında, veri bütünlüğünü ve ilişkisel tutarlılığı sağlamak amacıyla SQL Server LocalDB tercih edilmiştir. Veri tabanı şeması, Code-First yaklaşımı ile tasarlanmıştır; tablolar C# sınıfları (Models) üzerinden Entity Framework Core aracılığıyla oluşturulmuştur.

3.3.1 Varlık Modelleri (Entity Models)

Veri tabanı 12 temel entity model üzerine kurgulanmıştır:

Tablo 4: Entity Model Yapısı

Model	Açıklama
ApplicationUser	Kullanıcı bilgileri (Identity genişletmesi)
Profile	Kullanıcı profil detayları
Institution	Kamu kurumları
Complaint	Şikayet/öneri kayıtları
ComplaintUpdate	Başvuru durum güncellemeleri
Media	Medya dosyaları (fotoğraf, video, ses)
Badge	Rozet tanımları
UserBadge	Kullanıcı-Rozet ilişkisi
Warning	Kullanıcı uyarıları
BannedUser	Yasaklı kullanıcılar
AuditLog	Denetim kayıtları

3.3.2 Veritabanı İlişkileri

- **ApplicationUser - Complaint:** Bir-Çok (Bir kullanıcı birden fazla şikayet oluşturabilir)
- **ApplicationUser - Profile:** Bir-Bir (Her kullanıcının tek profili vardır)
- **ApplicationUser - UserBadge:** Bir-Çok (Bir kullanıcı birden fazla rozet kazanabilir)
- **Complaint - Media:** Bir-Çok (Bir şikayete birden fazla medya eklenebilir)
- **Complaint - ComplaintUpdate:** Bir-Çok (Bir şikayetin birden fazla güncelleme kaydı olabilir)
- **Institution - Complaint:** Bir-Çok (Bir kuruma birden fazla şikayet iletilebilir)

3.3.3 DbContext Yapılandırması

ApplicationDbContext sınıfı, tüm DbSet tanımlamalarını ve model konfigürasyonlarını içerir:

- Profiles, Institutions, Complaints, ComplaintUpdates
- MediaFiles, Badges, UserBadges
- Warnings, BannedUsers, AuditLogs

3.4 Kullanıcı Kimlik Doğrulama

Sistemin güvenliği, ASP.NET Core Identity altyapısı ile sağlanmaktadır. Modern güvenlik standartlarına uygun olarak çeşitli koruma mekanizmaları uygulanmıştır.

3.4.1 Şifre Politikası

- Minimum 8 karakter uzunluk
- En az bir büyük harf
- En az bir küçük harf
- En az bir rakam
- En az bir özel karakter

3.4.2 Hesap Kilitleme

- 5 başarısız giriş, denemesi - 10 dakika hesap kilitleme
- Yeni kullanıcılar için kilitleme aktif

3.4.3 Rol Tabanlı Yetkilendirme

Bes, farklı rol için authorization policy'leri tanımlanmıştır:

- RequireCitizenRole
- RequireInstitutionRole
- RequireModeratorRole
- RequireAdminRole
- RequireModeratorOrAdmin

3.4.4 Oturum Yönetimi

- Cookie tabanlı kimlik doğrulama
- 14 gün oturum süresi
- Sliding expiration aktif
- Özel yönlendirme sayfaları (Login, Logout, AccessDenied)

4 DEĞERLENDİRME

Bu bölümde; Kamu Köprüsü projesinin kullanıcı deneyimi (UX) ve kullanıcı arayüzü (UI) açısından basarımı, teknik altyapısının sağladığı avantajlar ve geliştirme sürecinde karşılaşılan kısıtlar ele alınmıştır.

4.1 Kullanılabilirlik Değerlendirmesi

Geliştirilen sistemin kullanılabilirliği, kullanıcıların platform üzerindeki görevlerini ne kadar etkin, verimli ve memnuniyet verici bir şekilde gerçekleştirebildiği üzerinden analiz edilmiştir.

- **Arayüz Tutarlılığı:** Razor Views ve Bootstrap CSS ile tüm sayfalarda görsel tutarlılık sağlanmıştır. Ortak layout yapısı sayesinde kullanıcılar platform içerisinde rahatça gezinebilmektedir.
- **Duyarlı Tasarım (Responsiveness):** Bootstrap grid sistemi kullanılarak masaüstü, tablet ve mobil cihazlarda sorunsuz çalışma sağlanmıştır.
- **Rol Bazlı Arayüzler:** Her kullanıcı rolü için özelleştirilmiş, dashboard ve menüler tasarlanmıştır. Kullanıcılar yalnızca yetkili oldukları işlevleri görmektedir.
- **Geri Bildirim Mekanizmaları:** Form gönderimi, durum değişiklikleri ve hatalar için anlık bilgilendirmeler sağlanmıştır. Türkçe hata mesajları kullanıcı deneyimini iyileştirmektedir.
- **Oyunlaştırma:** Puan, seviye ve rozet sistemi kullanıcı motivasyonunu artırmaktadır. Dashboard'da ilerleme durumu görsel olarak sunulmaktadır.

4.2 Teknik Değerlendirme

- **Güçlü Yönler:**
 - Güncel teknoloji yığını (.NET 10.0, EF Core 10.0)
 - Modüler ve katmanlı mimari
 - Kapsamlı güvenlik önlemleri
 - RESTful API ve Swagger dokümantasyonu

- Ölçeklenebilir veritabanı tasarımı

- **Kısıtlar ve Gelecek Geliştirmeler:**

- E-posta/SMS bildirimleri henüz placeholder durumunda
- SignalR gerçek zamanlı bildirimler gelecek sürümde planlanmaktadır
- Mobil uygulama API hazır ancak mobil istemci geliştirilmemiştir
- Dosya depolama yerel sunucuda; Azure/AWS entegrasyonu opsiyoneldir

5 SONUÇ

Bu çalışma kapsamında; vatandaşların kamu kurumlarıyla iletişimini dijitalleştirerek, basıyuru süreçlerini şeffaflaştırmak ve kullanıcı katılımını teşvik etmek amacıyla geliştirilen "Kamu Köprüsü" projesi başarıyla tamamlanmıştır. Proje, sadece bir yazılım uygulaması olarak değil, aynı zamanda e-devlet vizyonuna katkıda bulunan bir dijital platform olarak tasarlanmıştır.

Teknik açıdan değerlendirildiğinde; projenin .NET 10.0 ve Entity Framework Core 10.0 gibi endüstrinin en güncel teknolojilerinin kullanılması, sistemin yüksek performanslı, güvenli ve sürdürülebilir bir mimariye sahip olmasını sağlamıştır. Benimsenen Katmanlı Mimari (Layered Architecture) yaklaşımı ve MVC tasarım deseni sayesinde sistem modüler bir yapıya kavuşturulmuş; bu da projenin gelecekteki geliştirmelere ve ölçeklendirmelere açık olmasını garanti altına almıştır.

Bes, farklı kullanıcı rolü (Vatandaş, Kurum Temsilcisi, STÖ Koordinatörü, Moderatör, Admin) için özelleştirilmiş, arayüzler, çoklu medya desteği (fotoğraf, video, ses), rozet ve seviye sistemi ile oyunlaştırma mekanizmaları, dört aşamalı moderasyon sistemi ve kapsamlı yönetici paneli ile bütünleşik bir çözüm sunulmuştur.

Güvenlik açısından ASP.NET Core Identity altyapısı, rol tabanlı yetkilendirme, hesap kilitleme mekanizmaları ve güvenli şifre politikaları ile kullanıcı verilerinin korunması sağlanmıştır. RESTful API yapısı ve Swagger dokümantasyonu ile sistemin üçüncü parti entegrasyonlara ve mobil uygulama geliştirmelerine açık olması garanti altına alınmıştır.

Sosyal açıdan ele alındığında; Kamu Köprüsü, vatandaş-kurum iletişimindeki kopukluğu gidermek, basıyuru süreçlerinde şeffaflık sağlamak ve dijital ortamda vatandaş katılımını teşvik etmek konusunda önemli bir adım atmıştır. Oyunlaştırma mekanizmaları ile kullanıcı motivasyonu artırılmış, rozet ve seviye sistemi ile topluluk bağlılığı güçlendirilmiştir.

Sonuç olarak; modern web teknolojilerinin e-devlet çözümleriyle bütünleştirildiği bu proje, hem yazılım mühendisliği standartlarını karşılayan teknik bir başarı örneği sergilemiş, hem de hedef kitlesinin ihtiyaçlarına doğrudan yanıt veren işlevsel ve toplumsal faydası yüksek bir çözüm olarak ortaya çıkmıştır.

6 KAYNAKLAR

Bu çalışmanın gerçekleştirilmesi sürecinde faydalanılan literatür, standartlar ve teknik kılavuzlar aşağıda sunulmuştur:

6.1 Uluslararası Standartlar

- IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society.
- IEEE Std 1016-2009. IEEE Standard for Information Technology—Systems Design—Software Design Descriptions. IEEE Computer Society.
- IEEE Std 829-2008. IEEE Standard for Software and System Test Documentation. IEEE Computer Society.

6.2 Teknik Dokümantasyon ve Kılavuzlar

- Microsoft. (2025). .NET Documentation. Microsoft Corporation.
<https://learn.microsoft.com/en-us/dotnet/>
- Microsoft. (2025). ASP.NET Core Documentation. Microsoft Corporation.
<https://learn.microsoft.com/en-us/aspnet/core/>
- Microsoft. (2025). Entity Framework Core Documentation. Microsoft Corporation.
<https://learn.microsoft.com/en-us/ef/core/>
- Microsoft. (2025). ASP.NET Core Identity Documentation. Microsoft Corporation.
<https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity>
- Swagger/OpenAPI. (2025). OpenAPI Specification. SmartBear Software.
<https://swagger.io/specification/>