

**ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



AĞ TABANLI TEKNOLOJİLER VE UYGULAMALARI

Kütüphane Yönetim Sistemi Projesi

PROJE TESLİM DOKÜMANI

Yasin Ovat

22290097

VIDEO URL :

https://drive.google.com/file/d/1FJ0rlxugqh9llghFq8BC8PDBLfSQYUW/view?usp=drive_link

NOT : Swagger dokümantasyonu yok çünkü proje .NET Core MVC mantığıyla geliştirilmiştir.

1. PROJE HAKKINDA

Proje Adı: KutuphaneMVC

Geliştirme Platformu: ASP.NET Core MVC (.NET 9.0)

Veritabanı: Microsoft SQL Server (Entity Framework Core)

Proje Türü: Web Tabanlı Kütüphane Yönetim Sistemi

Bu proje, geleneksel kütüphane işlemlerini dijitalleştirerek bir web uygulamasıdır. Hem kütüphane personelinin (admin) işlerini kolaylaştırmayı hem de kütüphane kullanıcılarının online olarak kitap taleplerinde bulunmasını sağlamayı hedefler. Projenin temel mantığı, fiziksel kütüphane süreçlerini dijital ortama taşıyarak daha hızlı, takip edilebilir ve organize bir sistem oluşturmaktır.

2. TASARIM KARARLARI VE MİMARİ MANTIK

2.1. Neden MVC Mimarisi?

Model-View-Controller yapısı seçilmiştir çünkü veri yönetimi (Model), kullanıcı arayüzü (View) ve iş mantığını (Controller) birbirinden ayırrı. Bu sayede kitap listesi görünümünde yapılacak bir değişiklik, veritabanı yapısını etkilemez veya iş mantığını bozmaz. Her katman bağımsız çalışır ve bakım kolaylaşır.

2.2. Proje Yapısının Mantığı

- Models klasörü:** Kütüphanenin temel varlıklarını temsil eder (Kitap nedir? Üye nedir? Ödünç işlemi nasıl saklanır?)
- Controllers klasörü:** Her varlık için ayrı controller, sorumlulukları ayırır (BookController sadece kitaplarla, UserManagementController sadece kullanıcılarla ilgilenir)
- Views klasörü:** Her controller için ayrı view klasörü, hangi sayfanın hangi işe ait olduğunu net gösterir
- Filters klasörü:** Güvenlik katmanı, her sayfada tek tek oturum kontrolü yazmak yerine merkezi bir yetkilendirme sistemi

3. ROL TABANLI ERİŞİM MANTIĞI

3.1. Admin Rolünün Mantığı

Kütüphane personeli tüm yetkilere sahip olmalıdır çünkü envanter yönetimi, üye kayıtları, kitap girişi gibi operasyonel işler onların sorumluluğundadır. Admin kullanıcıları sisteme veri girer, kullanıcı taleplerini değerlendirir ve kütüphane politikalarını uygular. Bu nedenle tüm CRUD işlemlerine ve onay mekanizmalarına erişimleri vardır.

3.2. User Rolünün Mantığı

Normal kullanıcılar kütüphaneye kayıtlı üyelerdir ve sadece kendi işlemlerini yönetebilmelidir. Başkalarının taleplerini görmemeleri gerekmektedir (gizlilik). Sadece okuma, talep oluşturma ve kendi taleplerini takip etme yetkileri vardır. Bu, hem güvenlik hem de kullanıcı deneyimi açısından önemlidir.

3.3. Session ile Yetkilendirme Mantığı

Her kullanıcı giriş yaptığında bilgileri session'da saklanır. Böylece her sayfa değişiminde "kim bu kullanıcı?" sorusuna hızlıca cevap verilebilir. Filter mekanizması sayesinde her sayfaya girmeden önce otomatik kontrol yapılır ve yetkisiz erişimler engellenir.

4. ANA MODÜLLER VE İŞLEYİŞ MANTIĞI

4.1. Kullanıcı-Üye İlişkisi

Projedeki en önemli tasarım kararlarından biri, User (sistem kullanıcısı) ile Member (kütüphane üyesi) ayrılmıdır.

Mantık: Gerçek hayatta bir kütüphane üyesi olmak ile sisteme giriş yapabilmek farklı şeylerdir. Bir admin personel sisteme girebilir ama kütüphane üyesi olmayabilir. Bir kütüphane üyesinin sisteme giriş yapabilmesi için önce User hesabı oluşturulmalıdır.

Çözüm: User tablosu kimlik doğrulama için, Member tablosu ise fiziksel kütüphane üyeliği için kullanılır. Normal kullanıcılar oluşturulduğunda hem User hem Member kaydı otomatik yapılır ve birbirine bağlanır. Adminler için sadece User kaydı yeterlidir.

4.2. Kitap Talep Sistemi – Dijital Onay Süreci

Problem: Geleneksel kütüphanelerde kullanıcı gişeye gider, kitap ister, personel kontrol eder ve verir. Dijital sistemde bu nasıl olmalı?

Çözüm - LoanRequest Modülü:

1. Kullanıcı online olarak kitap talebinde bulunur (fiziksel gelmeye gerek yok)
2. Talep "Beklemede" durumunda sisteme kaydedilir
3. Admin panelinde bekleyen talepler listelenir
4. Admin değerlendirir: stok var mı, kullanıcı güvenilir mi, kitap teslim edilebilir mi?
5. Onaylarsa otomatik olarak ödünç kaydı oluşur, reddederse kullanıcı bilgilendirilir

Avantajları:

- Kullanıcı fiziksel olarak kütüphaneye gelmeden talepte bulunabilir
- Tüm talepler kayıt altında, takip edilebilir
- Personel zamanında talepleri değerlendirebilir (mesai dışı gelen talep masaide işlenir)
- İstatistik tutulabilir (hangi kitaplar çok isteniyor?)

4.3. Stok Yönetimi ve Otomatik Güncelleme

Mantık: Bir kitap ödünç verildiğinde stok azalmalı, iade edildiğinde artmalıdır. Bu manuel yapılırsa hata riski yüksektir.

Çözüm: Talep onaylanırken sistem otomatik olarak stok kontrolü yapar. Stok varsa onaylar ve stoğu 1 azaltır. İade işleminde tersini yapar. Tüm bu işlemler transaction ile yapılır, yani ya hepsi başarılı olur ya hiçbiri (yarım işlem olmaz).

4.4. Transaction Kullanımının Mantığı

Örnek senaryo: Admin bir talebi onayladı, Loan kaydı oluşturuldu ama tam o sırada elektrik gitti ve stok güncellenemedi. Sistemde kitap ödünç verilmiş gözükür ama stok düşmemiştir. İlerleye stok fazla gösterir, kitap aslında rafda yoktur.

Çözüm: Transaction tüm adımları bir paket olarak görür. Herhangi bir adımda hata olursa tümü geri alınır. Bu sayede veri tutarlılığı sağlanır.

5. GÜVENLİK MANTIĞI

5.1. Şifre Saklama Mantığı

Şifreler asla açık metin olarak saklanmaz. BCrypt ile hashlenmiş hali veritabanına kaydedilir. Giriş sırasında kullanıcının girdiği şifre hashlenir ve veritabanındaki hash ile karşılaştırılır. Böylece veritabanı çalınsa bile şifreler okunamaz.

5.2. Cascade Delete Engellemesi

Eğer bir kullanıcı silinirse, o kullanıcının oluşturduğu tüm talepler de silinirse tarihçe kaybolur. Bu yüzden silme işlemleri kısıtlanmıştır (Restrict). Kullanıcı pasif hale getirilir (IsActive=false) ama verisi saklanır.

5.3. Filter Sistemi ile Merkezi Güvenlik

Her controller'da "kullanıcı giriş yaptı mı?" kontrolü yazmak yerine, AuthFilter tüm sayfalara otomatik uygulanır. AuthorizeRole attribute'u ile rol kontrolü de yapılır. Merkezi sistem, güvenlik açığı riskini azaltır.

6. VERİTABANI TASARIM MANTIĞI

6.1. 5 Ana Tablo ve Neden Böyle?

Users Tablosu: Sisteme giriş yapabilecek herkes. Username, hash'lenmiş şifre, rol bilgisi içerir.

Members Tablosu: Kütüphane üyeleri. Ad-soyad, iletişim bilgileri. User'dan ayrı tutulması admin kullanıcılarının üye olmamasına imkan tanır.

Books Tablosu: Kitap envanteri. Başlık, yazar, tür, stok miktarı. Google Books API ile görsel desteği eklenmiştir (kullanıcı deneyimi).

LoanRequests Tablosu: Talep geçmişi. Kim, hangi kitabı, ne zaman talep etmiş, durum ne? Bu tablo fiziksel kütüphanelerde olmayan ama dijital sisteme çok değerli bir kayıttır.

Loans Tablosu: Gerçekleşmiş ödünç işlemleri. Alış tarihi, iade tarihi, son teslim tarihi. Gecikme kontrolü için kullanılır.

İlişkilerin Mantığı

- Bir talep bir kullanıcıya aittir ama bir admin tarafından işlenir (iki farklı User ilişkisi).
- Bir talep bir kitap ve bir üye için yapılır (Book ve Member ilişkileri).
- Bu ilişkiler sayesinde "Bu kitap kimin tarafından ne zaman talep edildi ve kim onayladı?" sorusuna eksiksiz cevap verilebilir.

7. KURULUM VE İLK ÇALIŞTIRMA

7.1. Otomatik Veritabanı Oluşturma Mantığı

Proje ilk çalıştırıldığında veritabanı yoksa otomatik oluşturulur. Migration dosyaları sayesinde tablo yapıları otomatik kurulur. Bu, projeyi farklı bir bilgisayara taşımayı kolaylaştırır.

7.2. Seed Admin Kullanıcısı

Sistem ilk kurulduğunda kimse giriş yapamaz çünkü hiç kullanıcı yoktur. Bu sorunu çözmek için otomatik olarak bir admin kullanıcısı oluşturulur (admin/admin123). İlk giriş bu hesapla yapılır ve ardından yeni kullanıcılar eklenebilir.

8. ÖNEMLİ İŞ AKIŞLARI

8.1. Yeni Kullanıcı Ekleme Akışı (Admin Tarafından)

- Admin yeni kullanıcı formu doldurur.
- Sistem username ve email'in benzersiz olup olmadığını kontrol eder (aynı kullanıcı adıyla iki kişi olamaz).
- Eğer rol "User" seçilmişse önce Member kaydı oluşturulur.
- Ardından User kaydı oluşturulur ve Member'a bağlanır.
- Şifre hashlenmiş olarak saklanır.
- Tüm işlem transaction ile güvenli yapılır.

Mantık: Normal kullanıcılar hem sisteme giriş yapabilmeli hem de kütüphane üyesi olmalıdır, bu nedenle iki kayıt birlikte oluşturulur.

8.2. Kitap Ödünç Alma Tam Döngüsü

1. Kullanıcı Tarafı: Kitap kataloğu arama yapar, beğenilen kitabı bulur, "Talep Et" butonuna basar
2. Sistem: LoanRequest kaydı oluşturur, durum "Pending", talep tarihi otomatik eklenir
3. Admin Tarafı: Bekleyen talepleri listede görür, kitap bilgisini ve talep eden kişiyi inceler
4. Onay Sırasında: Sistem stok kontrolü yapar (stok 0 ise uyarı verir)
5. Onay Sonrası: Loan kaydı oluşur, stok 1 azalır, talep durumu "Approved" olur, admin bilgisi kaydedilir
6. İade Sırasında (opsiyonel): Loan kaydına iade tarihi eklenir, stok tekrar artırılır

Mantık: Tüm süreç kayıt altındadır. Kim, ne zaman, hangi kitabı talep etmiş, kim onaylamış, ne zaman iade etmiş - hepsi takip edilebilir.

9. TEST SONUÇLARI VE DOĞRULAMA

Proje teslim öncesi tüm kritik işlevler manuel olarak test edilmiş ve başarıyla çalıştığı doğrulanmıştır.

9.1. Kimlik Doğrulama ve Yetkilendirme Testleri

Admin girişi: admin/admin123 bilgileriyle giriş yapıldı, ana sayfaya yönlendirme çalışıyor.

Yanlış şifre kontrolü: Hatalı şifre girildiğinde hata mesajı gösteriliyor, giriş yapılmıyor.

Session kontrolü: Oturum açmadan admin sayfalarına erişim denendiğinde login sayfasına yönlendirme çalışıyor.

Rol bazlı erişim: Normal kullanıcı ile admin sayfalarına erişim engellendiği doğrulandı.

Çıkış işlemi: Logout butonuna basıldığında session temizleniyor ve login sayfasına dönüş yapılıyor.

9.2. Kullanıcı Yönetimi Testleri

Yeni kullanıcı oluşturma: Admin hesabı ile yeni kullanıcı eklendi, hem User hem Member kaydı otomatik oluşturuldu.

Aynı kullanıcı adı kontrolü: Mevcut bir username ile kayıt denendiğinde uyarı mesajı gösteriliyor.

Email doğrulama: Aynı email adresiyle tekrar kayıt yapılamıyor, hata mesajı çalışıyor.

Kullanıcı düzenleme: Mevcut kullanıcı bilgileri güncellendiğinde değişiklikler veritabanına kaydediliyor.

Kullanıcı pasif etme: IsActive=false yapılan kullanıcı giriş yapamıyor, sistem engelliyor.

9.3. Kitap Yönetimi Testleri

Kitap ekleme: Yeni kitap formu dolduruldu, kitap başarıyla veritabanına eklendi, liste sayfasında görüntüleniyor.

Kitap düzenleme: Mevcut kitap bilgileri (başlık, yazar, stok) güncellendiğinde değişiklikler kaydediliyor.

Kitap silme: Kitap silme işlemi çalışıyor, veritabanından kayıt kaldırılıyor.

Stok kontrolü: Negatif stok girişi engellendiği, validation mesajının gösterildiği doğrulandı.

Google Books görseli: Kitap oluşturulurken imageUrl alanı girildiğinde görsel liste sayfasında gösteriliyor.

9.4. Üye Yönetimi Testleri

Üye ekleme: Yeni üye formu ile ad-soyad, telefon ve email bilgileri kaydediliyor.

Üye düzenleme: Üye bilgileri güncellendiğinde değişiklikler başarıyla kaydediliyor.

Üye silme: Üye silme işlemi çalışıyor.

Email format kontrolü: Geçersiz email formatı girildiğinde validation hatası gösteriliyor.

9.5. Kitap Talep ve Onay Sistemi Testleri

Talep oluşturma: Normal kullanıcı ile kitap talebi oluşturuldu, durum "Pending" olarak kaydedildi.

Talep listeleme: Admin panelinde bekleyen talepler filtreyle gösteriliyor.

Stokta olmayan kitap talebi: Stok=0 olan kitap için talep onaylanmaya çalışıldığından uyarı mesajı gösteriliyor.

Talep onaylama: Admin tarafından onaylanan talep "Approved" durumuna geçiyor.

Otomatik Loan oluşturma: Onaylanan talep için Loan kaydı otomatik oluşturuluyor.

Stok güncelleme: Talep onaylandığında kitap stoğu otomatik 1 azalıyor.

Transaction kontrolü: Onay sırasında hata oluştuğunda (stok yetersiz) hiçbir işlem gerçekleşmiyor, veri tutarlığı korunuyor.

Talep reddetme: Admin tarafından reddedilen talep "Rejected" durumuna geçiyor, stok değişmiyor.

9.6. Ödünç İşlem Testleri

Ödünç kaydı görüntüleme: Admin panelinde tüm ödünç kayıtları listeleniyor.

Manuel ödünç oluşturma: Admin tarafından manuel Loan kaydı oluşturulabiliyor.

İade tarihi güncelleme: Ödünç kayıtlarında iade tarihi eklendiğinde kaydediliyor.

Tarih kontrolleri: Alış tarihi, iade tarihi ve son teslim tarihi doğru şekilde saklanıyor.

9.7. Veritabanı ve Veri Bütünlüğü Testleri

Migration çalışması: Veritabanı migration dosyaları başarıyla uygulandı, tüm tablolar oluşturuldu

Foreign key ilişkileri: User-Member, LoanRequest-User, LoanRequest-Book ilişkileri doğru çalışıyor

Cascade delete koruması: Kullanıcı silinmek istendiğinde ilişkili talepleri varsa işlem engelleniyor

Auto seed admin: İlk çalışmada admin kullanıcıyı otomatik oluşturuyor

9.8. Kullanıcı Arayüzü ve Navigasyon Testleri

Menü navigasyonu: Admin ve User için farklı menüler gösteriliyor

Buton ve linkler: Tüm CRUD işlemleri için butonlar çalışıyor (Ekle, Düzenle, Sil, Detay)

TempData mesajları: Başarılı işlem sonrası bilgilendirme mesajları gösteriliyor

Hata mesajları: Validation hataları, stok uyarıları ve diğer hatalar kullanıcıya gösteriliyor

Responsive tasarım: Sayfalar farklı ekran boyutlarında düzgün görüntüleniyor

9.9. Performans ve Güvenlik Testleri

Session timeout: 30 dakika boyunca işlem yapılmadığında oturum sonlanıyor

BCrypt şifreleme: Veritabanında şifreler hash'li olarak saklanıyor, açık metin görünmüyor

CSRF koruması: POST işlemlerinde anti-forgery token kontrolü çalışıyor

SQL Injection koruması: Entity Framework parametreli sorguları kullandığı için SQL injection riski yok

Proje kapsamı dökümanında belirtilen tüm eksikler tamamlanmıştır (Kullanıcı kısmı eklenmiş ve VT tablo sayısı 5'e yükseltilmiştir.).