

Adalet Kapısı

line 1: 1st Ali Buğra Eroğlu
Numara: 231307114

line 1: 2nd M. Taha Kazıkoğlu
Numara: 241307121

line 1: 3rd Ömer Faruk Güler
Numara: 241307114

Abstract— Bu proje, hukuk bürolarının dava süreçlerini dijital ortama taşımak ve işlemleri otomatik hale getirmek amacıyla geliştirilmiştir. Sistem; avukatlar ve müvekkiller arasında yürütülen dava takibini kolaylaştırmak, duruşma bilgilerini düzenli bir şekilde tutmak ve güncellemelerde taraflara otomatik bildirim göndermek gibi işlevler sunar. Projenin ön yüzü (frontend), kullanıcı dostu ve sade bir arayüz oluşturmak amacıyla HTML ve CSS kullanılarak geliştirilmiştir. Arka plan (backend) işlemleri ise .NET altyapısı ile yazılmış; veritabanı işlemleri, iş kuralları ve tetikleyiciler (trigger) bu katmanda yönetilmiştir. Veriler, MySQL tabanlı ilişkisel bir veritabanında saklanmakta; view ve trigger gibi araçlar kullanılarak veri işlemleri otomatikleştirilmiştir. Genel olarak sistem, avukat ve müvekkil rollerine göre yetkilendirme yapısı sunarak hem kullanıcı deneyimini iyileştirmiş hem de veri güvenliğini sağlamıştır. Bu sayede hukuk büroları için iş yükünü azaltan, organize ve kullanıcı dostu bir çözüm ortaya konmuştur.

Keywords— Hukuk Otomasyonu, Dava Takip Sistemi, MySQL, .NET, HTML, Trigger, View, Rol Yönetimi

I. GİRİŞ

Bu proje, hukuk firmaları için yasal süreçleri dijitalleştirmek ve otomatikleştirmek için geliştirilmiştir. Öncelikli hedef, davaların, duruşmaların, müvekkillerin ve iç operasyonların yönetimini kolaylaştıran ve kullanıcı dostu bir şekilde verimli bir sistem sağlamaktır. Sistem, hukuk büroları ve müvekkilleri (davacılar ve davalılar) tarafından kullanılmak üzere tasarlanmıştır ve her iki tarafın da yasal prosedürleri daha şeffaf ve sistematik bir şekilde takip etmesini sağlar.

Projenin temel işlevleri şunlardır:

- Hukuki davalar hakkında detaylı bilgilerin kaydedilmesi ve yönetilmesi (davalar).
- Avukatlar (avukatlar) ve müvekkiller (müvekkiller) için yapılandırılmış profillerin tutulması.
- Yasal süreç boyunca duruşmaların ve ilgili aşamalarının izlenmesi.
- Bir davada güncellemeler meydana geldiğinde (örneğin, aşama değişikliği, tarih değişikliği) otomatik olarak bildirim (bildirimler) oluşturma.

Teknik açıdan, sistemin ön yüzü HTML ve CSS kullanılarak geliştirildi ve basit ve erişilebilir bir kullanıcı arayüzü sundu. Arka uç, iş mantığını ve veritabanı işlemlerini yürüten .NET kullanılarak uygulanmıştır.

Uygulamanın veri yapısını desteklemek için MySQL tabanlı ilişkisel bir veritabanı tasarlanmıştır. Karmaşık sorguları basitleştirmek ve kullanıcı rollerini mantıksal olarak ayırmak için görünüm (view'ler) aktif olarak kullanıldı (örneğin, bir avukatın müvekkillerini veya bir müvekkilin avukatlarını listelemek). Ayrıca, kalan ödemelerin hesaplanması, zamanlama çakışmalarının önlenmesi ve bildirimlerin zamanında oluşturulmasının sağlanması gibi

kritik veritabanı işlemlerini otomatikleştirmek için tetikleyiciler uygulanmıştır. Tetikleyicilerde görünümün kullanılması, temiz ve bakımı yapılabilir bir mantık elde edilmesini sağladı.

Genel olarak bu proje, kullanılabilirlik, veri bütünlüğü ve otomasyona odaklanarak hukuk profesyonelleri için özel olarak tasarlanmış akıllı ve yarı otomatik bir sistem oluşturmaya yönelik temel bir adım niteliğindedir.

II. KULLANIM KOLAYLIĞI

Bu proje, hukuk bürolarının dava süreçlerini dijital ortama taşımak ve işlemleri otomatik hale getirmek amacıyla geliştirilmiştir. Sistem; avukatlar ve müvekkiller arasında yürütülen dava takibini kolaylaştırmak, duruşma bilgilerini düzenli bir şekilde tutmak ve güncellemelerde taraflara otomatik bildirim göndermek gibi işlevler sunar.

Projenin ön yüzü (frontend), kullanıcı dostu ve sade bir arayüz oluşturmak amacıyla HTML ve CSS kullanılarak geliştirilmiştir. Arka plan (backend) işlemleri ise .NET altyapısı ile yazılmış; veritabanı işlemleri, iş kuralları ve tetikleyiciler (trigger) bu katmanda yönetilmiştir. Veriler, MySQL tabanlı ilişkisel bir veritabanında saklanmakta; view ve trigger gibi araçlar kullanılarak veri işlemleri otomatikleştirilmiştir.

Genel olarak sistem, avukat ve müvekkil rollerine göre yetkilendirme yapısı sunarak iyileştirmiş kullanıcı deneyimini sağlamıştır. Bu sayede hukuk büroları için iş yükünü azaltan, organize ve kullanıcı dostu bir çözüm ortaya konmuştur.

III. GENEL SİSTEM MIMARISI

Bu projede Frontend-Backend-Veritabanı üçlüsü kullanılarak modern bir yazılım mimarisi uygulanmıştır. Her katman kendi görevinden sorumludur ve sistemin bütünlüğü bu yapıyla sağlanmıştır:

• Frontend (Ön Yüz)

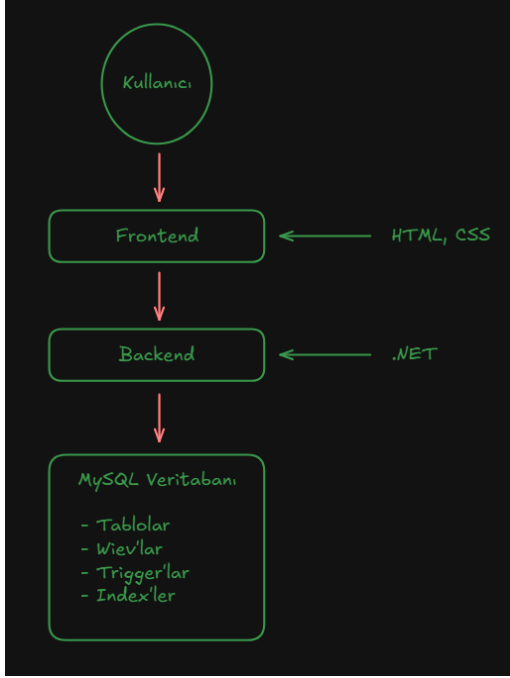
Kullanıcıların sistemi kolayca kullanabilmeleri için HTML ve CSS ile basit, anlaşılır ve erişilebilir bir arayüz geliştirilmiştir. Bu arayüz üzerinden dava bilgileri girilir, duruşmalar görüntülenir ve kullanıcıya sistemden gelen bildirimler gösterilir.

• Backend (Sunucu Katmanı)

.NET altyapısı kullanılarak geliştirilen bu katman; sistemdeki iş kurallarını yürütür, kullanıcı işlemlerini kontrol eder ve veritabanıyla haberleşir. Aynı zamanda trigger tetiklemeleri ve bildirim sistemleri gibi arka plan işlemleri de burada yönetilir.

- Veritabanı (MySQL)

Veriler MySQL ile tasarlanmış ilişkisel bir veritabanında tutulur. Kullanıcı bilgileri, davalar, duruşmalar, roller, bildirimler gibi tüm veri yapıları burada saklanır. View ve trigger kullanımı sayesinde veritabanı içinde birçok işlem otomatikleştirilmiştir.



Şekil 1. Genel Sistem Mimarisi Şeması

Öncelikle, sistemin hukuk büroları ve müvekkillerin kullanımına uygun, sezgisel bir tasarıma sahip olduğu teyit edilmiştir. Arayüz, farklı kullanıcı rolleri (avukat, müvekkil) için özelleştirilmiş ve her bir rolün ihtiyaç duyduğu temel işlemlere kolay erişim sağlayacak şekilde yapılandırılmıştır.

Örneğin, avukatlar dava yönetimi ve duruşma takvimine hızlıca ulaşabilirken, müvekkiller kendi davalarının özet bilgilerini ve önemli güncellemeleri net bir şekilde görüntüleyebilirler.

A. Veri Tabanı (MySQL)

- **Tarih ve saat** bilgileri: DATE, DATETIME ve TIME formatlarında işlenmektedir.
- **Para birimi** olarak **Türk Lirası (₺)** baz alınmış ve DECIMAL(10,2) türüyle kaydedilmiştir.
- **Kimlik numarası**, telefon gibi bilgiler VARCHAR tipinde saklanmakta ve benzersiz olacak şekilde yapılandırılmıştır.
- Süreç adımları (örneğin dava aşamaları), sıralı biçimde asama tablosunda numaralandırılmıştır.

1. Veritabanı Tabloları ve Amaçları

- **kisiler** tablosu, sistemdeki tüm kişilerin (avukat, müvekkil vb.) temel bilgilerini saklar. Temel alanları: Kisi_ID, Ad, Soyad, TC_Kimlik_No, E_posta, Telefon, Sifre, Baro_No, Adres_ID.rol / **kisi_rol**: Her kullanıcının rolünü (avukat, müvekkil, admin) tanımlar ve ilişkilendirir.

- **rol** tablosu, sistem kullanıcılarının rollerini tanımlar (avukat, müvekkil, admin). Temel alanları: Rol_ID, Rol.dava_taraf: Davaya katılan kişi ve rollerini belirtir (avukat veya müvekkil).
- **adres_avukat** tablosu, avukatların adres bilgilerini saklar. Temel alanları: Adres_ID, İl, İlçe, Adres.
- **dava** tablosu, tüm davaların kayıtlarını tutar. Temel alanları: Dava_ID, Dava_Konusu, Acilis_Tarihi, Guncelleme_Tarihi, Aciklama ve diğer tablolara referanslar içerir.
- **dava_taraf** tablosu, davalar ile kişiler (avukat/müvekkil) arasındaki ilişkiyi tanımlar. Temel alanları: Dava_ID, Kisi_ID, Rol_ID.
- **durusmalar** tablosu, duruşma kayıtlarını içerir. Temel alanları: Durusma_ID, Tarih, Saat, Dava_ID, Mahkeme_ID, DurusmaDurum_ID.
- **mahkemeler** tablosu, mahkeme bilgilerini saklar. Temel alanları: Mahkeme_ID, Ad, Şehir.
- **dava_durum** tablosu, davaların durumlarını tanımlar. Temel alanları: DavaDurum_ID, Dava_Turu.
- **taraf** tablosu, dava taraflarını tanımlar (davacı, davalı, tek taraf). Temel alanları: Taraf_ID, Taraf.
- **asama** tablosu, dava aşamalarını tanımlar. Temel alanları: Asama_ID, Asama.
- **durusma_durum** tablosu, duruşma durumlarını tanımlar. Temel alanları: DurusmaDurum_ID, Durusma_Turu.
- **odeme** tablosu, davalarla ilgili ödeme bilgilerini saklar. Temel alanları: Odeme_ID, Toplam_Miktar, Odenen_Miktar, Kalan_Miktar.
- **dava_tarihce** tablosu, dava ile ilgili yapılan değişikliklerin geçmişini tutar. Temel alanları: Tarihce_ID, Dava_ID, Asama, Guncelleme_Tarihi, Guncelleyen_ID.

2. İlişkiler ve Amaçları

- Kişi-Rol ilişkisi (N:M): Bir kişi birden fazla role sahip olabilir (örneğin hem avukat hem müvekkil).
- Kişi-Adres ilişkisi (1:N): Bir kişinin birden fazla adresi olabilir.
- Dava-Kişi ilişkisi (N:M): Bir dava birden fazla kişiyle ilişkili olabilir (avukatlar, müvekkiller).
- Dava-Duruşma ilişkisi (1:N): Bir davanın birden fazla duruşması olabilir.
- Dava-Mahkeme ilişkisi (N:1): Bir dava yalnızca bir mahkemede görülür.
- Dava-Ödeme ilişkisi (1:1): Her davanın bir ödeme kaydı vardır.

3. View (Görünümler) ve Amaçları

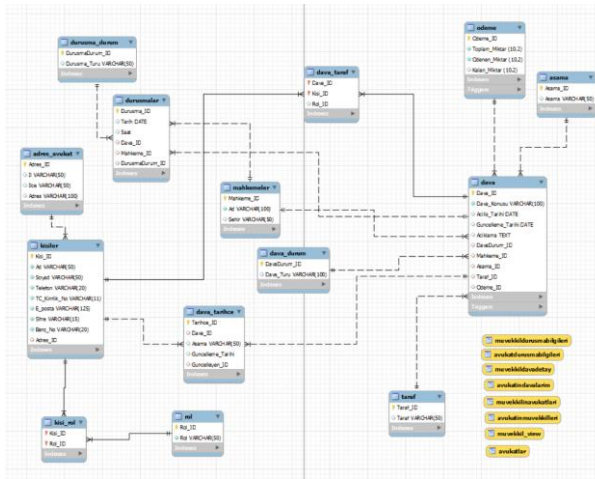
Sistemdeki karmaşık sorguları basitleştirmek için aşağıdaki view'ları kullandık.

- muvekkil_view: Müvekkillerin dava bilgilerini listeler.
- avukatlar: Dava bazında avukat bilgilerini listeler.
- MuvekkilinAvukatları: Müvekkillerin kendilerine atanmış avukatları görmesini sağlar.
- AvukatinMuvekkilleri: Avukatların müvekkillerini listelemelerini sağlar.
- AvukatinDavalarım: Avukatların rol aldığı davaları listeler.
- MuvekkilDavaDetay: Müvekkillerin dava detaylarını görmesini sağlar.
- AvukatDurusmaBilgileri: Avukatlar için duruşma bilgilerini listeler.
- MuvekkilDurusmaBilgileri: Müvekkiller için duruşma bilgilerini listeler.

4. Trigger (Tetikleyiciler) ve Amaçları

Sistemdeki veri tutarlılığını sağlamak için aşağıdaki tetikleyicileri kullandık.

- trg_odeme_kalan_hesapla: Ödeme kaydı oluşturulduğunda kalan miktar otomatik olarak hesaplanır.
- trg_odeme_kalan_guncelle: Ödeme kaydı güncellendiğinde kalan miktar otomatik güncellenir.
- tg_dava_durum_guncelle_update: Dava aşaması değiştiğinde durum otomatik güncellenir.
- tg_dava_durum_guncelle_insert: Yeni dava eklendiğinde durum otomatik atanır.
- otomatik_guncelleme_tarihi: Dava güncellendiğinde güncelleme tarihi otomatik olarak ayarlanır.



Şekil 2. Adalet Kapısı EER Diagramı

4. Indexler ve Amaçları

Sorgu performansını artırmak için aşağıdaki indeksler aşağıdaki gibi tanımlanmıştır:

- kisiler_Arama: TC kimlik, telefon ve e-posta ile kişi aramaları için birleşik indeks.
- kisiler_Avukat: Baro numarası, ad ve soyad ile avukat aramaları için birleşik indeks.
- dava_Mahkeme: Belirli bir mahkemeye ait davaları listelemek için.
- dava_Mahkeme: Belirli bir mahkemeye ait davaları listelemek için.
- dava_taraf_DavaRol: Belirli bir davadaki avukat ve müvekkilleri hızlı sorgulamak için.
- dava_Durum_Asama: Dava durumu ve aşamasına göre arama yapmak için.

B. Backend Yapılandırması

1. Model Sınıfları

- Sistem, veritabanı tablolarını temsil eden aşağıdaki model sınıflarını kullanmaktadır:
 - 1) **Dava.cs**: Dava kayıtlarını temsil eder.
 - 2) **Muvekkil.cs**: Müvekkil bilgilerini temsil eder.
 - 3) **Avukat.cs**: Avukat bilgilerini temsil eder.
 - 4) **Durusma.cs**: Duruşma kayıtlarını temsil eder.
 - 5) **Mahkeme.cs**: Mahkeme bilgilerini temsil eder.
 - 6) **Kisiler.cs**: Kişi kayıtlarını temsil eder.

2. WiewModel Sınıfları

- Kullanıcı arayüzüne veri taşımak için aşağıdaki ViewModel sınıfları kullanılmaktadır:
 - 1) **DavaDetayViewModel**: Dava detaylarını görüntülemek için
 - 2) **DavaEkleViewModel**: Yeni dava eklemek için
 - 3) **DurusmalarViewModel**: Duruşma listelerini görüntülemek için
 - 4) **MuvekkilViewModel**: Müvekkil bilgilerini görüntülemek için
 - 5) **AvukatViewModel**: Avukat bilgilerini görüntülemek için

3. Controller Sınıfları

- Sistem, kullanıcı isteklerini işlemek için aşağıdaki controller sınıflarını kullanır:
 - 1) **AvukatLoginController**: Avukat girişi ve yönetim paneli işlevleri
 - 2) **MuvekkilLoginController**: Müvekkil girişi ve dava takibi işlevleri
 - 3) **LoginController**: Genel giriş ve kimlik doğrulama işlevleri

4. Transaction (İşlem) Yönetimi

Tüm ödeme, dava ve taraf ekleme adımları tek bir transaction bloğunda yürütülür. Adımlardan biri başarısız olursa ROLLBACK ile tüm değişiklikler geri alınır; hepsi sorunsuz tamamlanırsa COMMIT ile kalıcı hale gelir. Bu sayede yarım kalmış veya tutarsız kayıtlar oluşmaz.

- **Atomiklik:** Ödeme, dava kaydı ve taraf ekleme adımları tek bir blokta (BEGIN...COMMIT) yürütülür.
- **Geri alma (Rollback):** Bloğun içindeki herhangi bir adım başarısız olursa, tüm öncesi otomatik olarak geri alınır.
- **Kalıcılık (Commit):** Tüm adımlar hatasız çalışırsa, COMMIT ile değişiklikler veritabanına kesin kaydedilir.

- 1) **Ödeme ekleme** → INSERT INTO ödeme (kodda ödemeCmd.ExecuteScalar())
- 2) **Dava ekleme** → INSERT INTO dava (kodda davaCmd.ExecuteScalar())
- 3) **Taraf ekleme** → INSERT INTO dava_taraf (müvekkiller ve avukat için döngü ve son komut)
- 4) **ROLLBACK** → transaction.Rollback() bir adım başarısız olursa tüm iletişimleri geri alır

5. Veri Erişim Katmanı

- 1) **KisilerData.cs:** Veritabanı işlemlerini gerçekleştiren ana sınıf.
- 2) **CRUD (Create, Read, Update, Delete)** operasyonları bu katman üzerinden gerçekleştirilir.

IV. VERİ BÜTÜNLÜĞÜ VE DOKÜMANTASYON STANDARTLARI

A. Spesifikasyonların Bütünlüğünün Korunması

Sistem, yasal süreçlerin ve ilgili bilgilerin tutarlı ve düzenli bir şekilde yönetilmesini sağlamak amacıyla belirli bir yapı üzerine inşa edilmiştir. Tüm dava bilgileri, müvekkil ve avukat profilleri, duruşma kayıtları önceden tanımlanmış alanlara kaydedilir. Bu yapı, kullanıcıların bilgi girişi sırasında standart bir formatı takip etmelerini ve böylece sistemdeki verilerin bütünlüğünün korunmasını sağlar. Örneğin, bir dava kaydı oluşturulurken, davanın türü, tarafları, açılış tarihi gibi temel bilgiler belirli alanlara girilmek zorundadır. Bu sayede, farklı davaların karşılaştırılması ve raporlanması kolaylaşır.

B. Kısaltmalar ve Akronimler

Sistem içerisinde kullanılan tüm kısaltmalar ve akronimler, ilk kez kullanıldıkları yerde açıkça tanımlanmıştır. Hukuk alanında yaygın olarak bilinen ve karışıklığa yol açmayacağı düşünülen bazı standart kısaltmalar (örneğin, T.C., vb.) tanımlanmayabilir. Ancak, sistemimize özgü veya daha az bilinen tüm kısaltmalar, kullanıcıların kolayca anlayabilmesi için metin içinde ilk geçtiği yerde tam olarak açıklanmıştır. Başlıklar ve temel menü öğelerinde zorunlu olmadıkça kısaltma kullanımından kaçınılmıştır.

V. SONUÇ

Bu raporda, Büro Yönetim Sistemi'nin backend ve veritabanı mimarisi detaylı olarak incelenmiştir. Sistem, hukuk bürolarının günlük operasyonel ihtiyaçlarını karşılayacak şekilde tasarlanmış olup, modern web teknolojileri ve veritabanı yapısı kullanılarak geliştirilmiştir. İleriki aşamalarda, önerilen iyileştirmeler uygulanarak sistemin performansı, güvenliği ve kullanılabilirliği daha da artırılabilir.