**1. Proje Tanımı ve Amaç**

**Proje Adı:**  
**ikCore – İnsan Kaynakları Yönetim Platformu**

**Proje Tanımı:**  
ikCore, şirketlerin insan kaynakları süreçlerini dijitalleştirerek merkezi bir panel üzerinden yönetmelerine olanak sağlayan, çok kiracılı (multi-tenant) yapıda geliştirilmiş kurumsal bir web uygulamasıdır. ASP.NET MVC (.NET Framework 4.8) teknolojisi kullanılarak geliştirilen bu sistem; kullanıcı dostu arayüzü, rol bazlı erişim mimarisi ve modüler yapısıyla, İK departmanlarının tüm temel ve ileri düzey işlemlerini dijital ortama taşımayı hedefler. Uygulama ilk aşamada yerel veritabanı (on-premise SQL Server) üzerinden çalışacak şekilde tasarlanmış olup, ilerleyen aşamalarda bulut altyapı desteği planlanmaktadır.

**Amaç:**  
ikCore’un temel amacı, insan kaynakları departmanlarının tüm işlemlerini (personel yönetimi, izin ve bordro süreçleri, vardiya planlaması, işe alım, evrak takibi vb.) tek bir sistem üzerinden, güvenli, izlenebilir ve esnek bir şekilde yönetebilmesini sağlamaktır. Manuel işlemleri ortadan kaldırarak zaman tasarrufu sağlamak, hata payını azaltmak ve yöneticilere stratejik kararlar için güçlü raporlama araçları sunmak hedeflenmektedir.

**Hedef Kullanıcılar:**

* Şirket İK Yöneticileri
* İK Personelleri
* Departman Müdürleri
* Firma Sahipleri / Üst Düzey Yöneticiler
* Tüm Şirket Çalışanları
* Firma Dışı Aday Kullanıcılar (İşe alım başvuru süreci)

**Temel Özellikler:**

* Her kullanıcı yeni bir firma oluşturabilir veya mevcut bir firmaya firma kodu (ID) ile katılabilir.
* Her firma yalnızca kendi personel, departman ve organizasyon yapısını yönetebilir.
* Rol bazlı yetkilendirme sistemiyle farklı kullanıcı profilleri için özel erişim izinleri tanımlanabilir.
* Kullanıcılar yalnızca yetkili oldukları modüllere erişebilir.
* Gerçek zamanlı bildirimler (SignalR), e-posta bildirimleri ve işlem geçmişi kayıtları desteklenir.
* Personel, izin, bordro, vardiya, performans, işe alım ve evrak yönetimi gibi tüm İK fonksiyonları sistemde modüler olarak yer alır.
* Firma yöneticileri yalnızca ihtiyaç duydukları modülleri aktif edebilir.

**Faydaları:**

* Yerel sunucu altyapısıyla hızlı ve güvenli veri yönetimi
* Kuruma özel yapılandırılabilir modüller
* Vardiya ve puantaj sistemi ile operasyonel takip
* Dijital bordro ve izin hesaplamaları ile zaman kazancı
* İşe alım süreçlerinin dijital yönetimi
* Evrakların merkezi olarak saklanması ve izlenebilirliği
* Kullanıcı ve işlem bazlı denetim (audit log) desteği
* Gelecekte bulut geçişine hazır, genişletilebilir mimari

**2. Genel Gereksinimler**

**2.1. Sistem Gereksinimleri**

* **Platform:** ASP.NET MVC (Microsoft .NET Framework 4.8)
* **Veritabanı:** Microsoft SQL Server
  + Uygulama ilk aşamada yerel veritabanı (on-premise) desteğiyle çalışacaktır.
  + Gelecekte Microsoft Azure SQL veya benzeri bulut servislerine geçiş planlanmaktadır.
* **Mimari:** N-Katmanlı mimari
  + **Katmanlar:** UI (Web), Business, DataAccess, Entities
  + Modüler servis yapısı ve genişletilebilir controller mimarisi
* **Kimlik Doğrulama:**
  + E-posta ile kayıt ve giriş
  + Token tabanlı şifre sıfırlama mekanizması
  + Rol bazlı yetkilendirme: Firma Sahibi (Admin), İK Yöneticisi, Departman Müdürü, Çalışan
  + 2-Faktörlü Kimlik Doğrulama (2FA) planlamaya açıktır, başlangıç sürümünde yer almayacaktır.
* **Dil Desteği:**
  + Arayüz ve sistem dili Türkçe olarak tasarlanacaktır. Çok dilli destek şimdilik planlanmamaktadır.
* **Bildirim Sistemi:**
  + E-posta bildirimleri (SMTP veya MailKit üzerinden)
  + Gerçek zamanlı sistem içi bildirim altyapısı (SignalR ile)
  + (Opsiyonel) SMS bildirim altyapısı ilerleyen sürümlerde değerlendirilebilir

**2.2. Çok Kiracılı (Multi-Tenant) Yapı**

* Uygulama, çok kiracılı mimariyi destekleyecek şekilde geliştirilmiştir.
* Her kullanıcı sisteme giriş yaptığında:
  + Ya yeni bir firma hesabı oluşturabilir
  + Ya da mevcut bir firmaya **firma ID** ile katılabilir
* Firma kimliği (CompanyId) sistemin tüm modüllerinde temel referans olarak kullanılacak ve her kullanıcı, yalnızca bağlı olduğu firmaya ait verilere erişebilecektir.
* Tüm modüller ve kullanıcılar firma bazlı izole edilmiştir. Her firma kendi departmanlarını, çalışanlarını, modüllerini ve izin kurallarını bağımsız olarak yönetebilir.

**2.3. Kullanıcı Rolleri ve Yetki Seviyeleri**

| **Rol** | **Açıklama** |
| --- | --- |
| **Firma Sahibi (Admin)** | Sistemde ilk kullanıcıdır. Firma kurar, tüm modüllere ve ayarlara erişim yetkisine sahiptir. |
| **İK Yöneticisi** | Personel işlemleri, izin yönetimi, bordro oluşturma, işe alım ve performans gibi süreçleri yönetir. |
| **Departman Müdürü** | Yalnızca kendi departmanındaki çalışanları yönetir, izin ve avans taleplerini onaylar. |
| **Çalışan** | Kendi bilgilerini görüntüler, izin/avans başvurusu yapabilir, bordrosunu ve evraklarını inceleyebilir. |

**2.4. Entegrasyon Planı (API ve Harici Servisler)**

* **Mernis Kimlik Doğrulama API (test ortamı)**  
  TC Kimlik numarası, ad-soyad ve doğum tarihi doğrulaması yapılabilir.
* **GİB Vergi Kimlik Doğrulama API**  
  Firma oluşturma sırasında vergi numarası ve unvan eşleşmesini kontrol etmek amacıyla kullanılabilir.
* **E-Posta Servisi**  
  SMTP ya da MailKit kütüphanesi üzerinden kayıt, şifre sıfırlama ve bildirim e-postaları gönderilecektir.
* **SGK Bildirim API’leri** (ileride entegre edilebilir)  
  Personel işe giriş/çıkış işlemlerinin doğrudan SGK sistemine bildirilmesi planlanmaktadır.

**2.5. Güvenlik Gereksinimleri**

* Tüm şifreler **BCrypt** veya benzeri modern hashing algoritmaları ile saklanacaktır.
* Web.config üzerinden yönetilen hassas veriler (connection string, API key vb.) şifreli tutulacaktır.
* Üretim ortamı için **SSL (HTTPS)** zorunlu olacak; tüm bağlantılar güvenli protokoller üzerinden gerçekleştirilecektir.
* Şifre kurtarma işlemleri token tabanlı olarak zaman sınırıyla gerçekleştirilecektir.
* Geliştirme sürecinde **brute-force koruması**, **reCAPTCHA** ve benzeri bot korumaları planlama dahilindedir.

**3. Modül Listesi ve Açıklamaları**

ikCore, insan kaynakları departmanlarının tüm temel ve ileri seviye süreçlerini yönetebilmesi amacıyla modüler bir yapıda tasarlanmıştır. Her bir modül bağımsız olarak aktif/pasif hale getirilebilir ve şirketlerin ihtiyaçlarına göre özelleştirilebilir. Modüller firma bazında yönetilir, sadece ilgili firmanın kullanıcıları tarafından erişilebilir.

**3.1. Temel Yönetim Modülleri**

**1. Firma Yönetimi Modülü**  
Kullanıcıların firma oluşturmasını veya bir firmaya katılmasını sağlar. Firma kimlik bilgileri (vergi numarası, unvan, adres) burada tanımlanır. Her firma sistemde benzersiz bir ID ile temsil edilir.

**2. Personel Yönetimi Modülü**  
Firma bünyesindeki tüm çalışanların kimlik, iletişim, görev, işe giriş/çıkış gibi bilgileri bu modülde yönetilir. Kullanıcı ve çalışan ilişkisi kurularak çok rollü yapı desteklenir. Personel geçmiş hareketleri izlenebilir durumdadır.

**3. Departman ve Pozisyon Yönetimi**  
Firmaya bağlı departmanların ve pozisyonların tanımlandığı modüldür. Her departmana sorumlu müdür atanabilir. Organizasyon şeması bu yapı üzerinden oluşturulur.

**4. Yetkilendirme ve Rol Yönetimi**  
Sistem genelinde kullanıcıların erişim hakları bu modül üzerinden düzenlenir. Rol bazlı erişim modeli uygulanır ve firma yöneticisi rol ataması gerçekleştirebilir.

**3.2. Süreç Yönetim Modülleri**

**5. İzin Yönetimi**  
Çalışanların yıllık, mazeret, hastalık ve ücretsiz izin taleplerini iletebildiği, yöneticilerin onay/ret işlemlerini gerçekleştirebildiği modüldür. İzin kotaları, geçmiş izinler ve onay akışı sistemde kayıt altına alınır.

**6. Maaş ve Bordro Modülü**  
Personellere ait maaş tanımları tutulur ve aylık bordrolar oluşturulur. SGK kesintileri, net maaş hesaplamaları ve bordro çıktıları bu modül tarafından yönetilir.

**7. Avans Talep Modülü**  
Çalışanların avans talebinde bulunabildiği, onay ve ödeme takibinin yapıldığı modüldür. Talep geçmişi kayıt altına alınır.

**8. Performans Değerlendirme Modülü**  
Departman müdürleri ve İK yöneticileri tarafından çalışanlara yönelik performans değerlendirmeleri yapılabilir. Değerlendirme kriterleri ve puanlama yapısı özelleştirilebilir.

**9. Raporlama ve İstatistik Modülü**  
Firmanın personel yapısı, izin dağılımı, bordro toplamları, pozisyon bazlı dağılımlar gibi çeşitli raporları sunan ve grafiksel analiz ekranları içeren modüldür. Çıktılar Excel ve PDF formatında alınabilir.

**10. Bildirim ve Mesajlaşma Modülü**  
Sistem içi olaylar için kullanıcıya gerçek zamanlı bildirim sunar. İzin onayı, bordro yayımlanması gibi durumlar bu yolla çalışanlara iletilir. Gelişmiş bildirim sistemleriyle (SignalR ve SMTP) entegredir.

**11. Sistem Hareket Geçmişi (Audit Log) Modülü**  
Tüm önemli işlemler sistemsel olarak kayıt altına alınır. Kim, ne zaman, hangi işlemi yaptı gibi bilgiler takip edilebilir. Güvenlik ve denetlenebilirlik açısından zorunlu modüllerdendir.

**12. Modül Yönetimi**  
Firma yöneticilerinin hangi modülleri aktif kullanmak istediklerini seçebildiği yapı. Özelleştirilebilir mimariye sahip olduğundan, her firmanın yalnızca ihtiyaç duyduğu modülleri kullanması sağlanır.

**3.3. Genişletilmiş Fonksiyonel Modüller**

**13. Vardiya ve Puantaj Modülü**  
Firma tarafından tanımlanan vardiyalara göre çalışanların görevli olduğu günler, saatler ve giriş-çıkış bilgileri kaydedilir. Puantaj takibi ve vardiya planlaması yapılabilir. Manuel ya da cihaz entegrasyonuna açıktır.

**14. İşe Alım ve Aday Takip Modülü**  
Firmaların yeni pozisyonlar için iş ilanı açabileceği, aday başvurularını takip edebileceği modüldür. Özgeçmiş yükleme, değerlendirme ve başvuru durumu yönetimi bu modül üzerinden yürütülür.

**15. Çalışan Dosya ve Evrak Yönetimi**  
Çalışanlara ait belgelerin (sözleşme, diploma, kimlik fotokopisi vb.) sistem üzerinde dijital olarak saklandığı ve yönetildiği modüldür. Evrak türüne göre kategorilendirme yapılabilir, yükleme yetkileri kontrol edilebilir.

**4. Kullanıcı Roller ve Yetkileri – Yetki Matrisi**

| **Modül / İşlem** | **Firma Sahibi (Admin)** | **İK Yöneticisi** | **Departman Müdürü** | **Çalışan** |
| --- | --- | --- | --- | --- |
| Firma oluşturma / düzenleme | ✔ | – | – | – |
| Firma personelini davet etme | ✔ | ✔ | – | – |
| Personel ekleme / düzenleme / çıkarma | ✔ | ✔ | – | – |
| Kendi bilgilerini düzenleme | ✔ | ✔ | ✔ | ✔ |
| Departman oluşturma / pozisyon atama | ✔ | ✔ | – | – |
| Departman personeli görüntüleme | ✔ | ✔ | ✔ | – |
| İzin talebi oluşturma | – | – | – | ✔ |
| İzin onayı | ✔ | ✔ | ✔ | – |
| İzin geçmişi ve raporlarını görüntüleme | ✔ | ✔ | ✔ | ✔ |
| Maaş ve bordro tanımlama | ✔ | ✔ | – | – |
| Kendi bordrosunu görüntüleme | ✔ | ✔ | ✔ | ✔ |
| Avans talebi oluşturma | – | – | – | ✔ |
| Avans onayı | ✔ | ✔ | ✔ | – |
| Performans değerlendirmesi yapma | – | ✔ | ✔ | – |
| Kendi performans geçmişini görüntüleme | ✔ | ✔ | ✔ | ✔ |
| Raporlama ve istatistikleri görüntüleme | ✔ | ✔ | ✔ | – |
| Kullanıcıya rol atama ve yetki düzenleme | ✔ | (kısıtlı) | – | – |
| Firma modüllerini aktifleştirme / devre dışı bırakma | ✔ | ✔ | – | – |
| Bildirim geçmişini görüntüleme | ✔ | ✔ | ✔ | ✔ |
| Sistem loglarını (audit trail) görüntüleme | ✔ | ✔ | – | – |
| Vardiya tanımlama ve atama | ✔ | ✔ | ✔ | – |
| Kendi vardiya planını görüntüleme | ✔ | ✔ | ✔ | ✔ |
| Giriş-çıkış (puantaj) kayıtlarını görüntüleme | ✔ | ✔ | ✔ | ✔ |
| İş ilanı oluşturma / düzenleme | ✔ | ✔ | – | – |
| Aday başvurularını görüntüleme / değerlendirme | ✔ | ✔ | – | – |
| Aday olarak başvuru yapma (firma dışı kullanıcı) | – | – | – | ✔ (\*) |
| Çalışan evrakı yükleme / düzenleme | ✔ | ✔ | – | – |
| Kendi evraklarını görüntüleme | ✔ | ✔ | ✔ | ✔ |

* ✔ : Yetki var
* – : Yetki yok
* (kısıtlı): Yalnızca kendi firmasındaki kullanıcılar için yetki vardır
* (\*) Aday kullanıcı, firma hesabı olmayan ve sisteme başvuru formu üzerinden giriş yapan kullanıcıdır

**5. Veritabanı Tasarımı**

**5.1. Temel Tablolar**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| Companies | Firma bilgileri |
| Users | Sistemdeki tüm kullanıcı hesapları |
| Roles | Sabit rol tanımları (Admin, İK, Müdür, Çalışan vb.) |
| UserRoles | Kullanıcıların firma bazlı rol ilişkileri |
| Employees | Çalışan bilgileri (kullanıcıya bağlı ya da sadece sistem içi kayıt olabilir) |
| Departments | Firmanın organizasyonel birimleri |
| Positions | Departmanlara bağlı görev ve pozisyon tanımları |

**5.2. Süreç Tabloları**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| LeaveRequests | İzin talepleri ve onay süreci |
| AdvanceRequests | Avans talepleri ve ödeme takibi |
| Payrolls | Bordro verileri (brüt, net maaş, kesintiler) |
| PerformanceReviews | Performans değerlendirme kayıtları |
| Notifications | Sistem içi bildirimler (SignalR, e-posta tetikleyicileri) |
| AuditLogs | Kullanıcı işlemlerinin kayıt altına alındığı denetim günlüğü |

**5.3. Genişletilmiş Modül Tabloları**

**Vardiya ve Puantaj Takibi**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| Shifts | Vardiya tanımları (örneğin: Sabah, Akşam, Gece) |
| EmployeeShifts | Çalışanın belirli bir tarihte atandığı vardiya bilgisi |
| AttendanceRecords | Giriş-çıkış saatleri ve puantaj kayıtları |

**İşe Alım Süreci**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| RecruitmentPosts | Firma tarafından yayınlanan açık pozisyonlar |
| JobApplications | Aday başvuruları ve başvuruya ait bilgiler (özgeçmiş dahil) |

**Çalışan Dosya Yönetimi**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| DocumentTypes | Evrak türleri (sözleşme, diploma vb.) |
| EmployeeDocuments | Her çalışana ait yüklenen evraklar |

**5.4. Yardımcı Tablolar**

| **Tablo Adı** | **Açıklama** |
| --- | --- |
| Modules | Sistemdeki tüm modül tanımları |
| CompanyModules | Her firmanın hangi modülleri aktif kullandığını gösterir |
| PasswordResetTokens | Şifre sıfırlama işlemleri için geçici token kayıtları |

**5.5. Anahtar İlişkiler**

* Users.CompanyId → Companies.Id
* UserRoles.UserId → Users.Id, UserRoles.RoleId → Roles.Id
* Employees.UserId → Users.Id
* Departments.CompanyId → Companies.Id
* Positions.DepartmentId → Departments.Id
* Employees.PositionId → Positions.Id
* LeaveRequests.EmployeeId → Employees.Id
* AdvanceRequests.EmployeeId → Employees.Id
* Payrolls.EmployeeId → Employees.Id
* PerformanceReviews.EmployeeId → Employees.Id
* EmployeeShifts.EmployeeId → Employees.Id
* EmployeeShifts.ShiftId → Shifts.Id
* AttendanceRecords.EmployeeId → Employees.Id
* JobApplications.PostId → RecruitmentPosts.Id
* EmployeeDocuments.EmployeeId → Employees.Id, EmployeeDocuments.DocumentTypeId → DocumentTypes.Id
* CompanyModules.CompanyId → Companies.Id, CompanyModules.ModuleId → Modules.Id

**5.6. Tasarım Notları**

* Tüm veriler CompanyId üzerinden filtrelenecek şekilde multi-tenant prensibi ile yapılandırılacaktır.
* UserRoles tablosu sayesinde bir kullanıcının aynı anda birden fazla firmada farklı rol alabilmesi sağlanacaktır.
* Evrak yönetimi, dosya sistemine veya blob storage sistemine entegre çalışabilir.
* AuditLogs tablosu uygulamanın denetlenebilirliğini sağlamak için zorunlu kılınacaktır.
* Modules ve CompanyModules tablosu ile modüler yapı dinamik olarak yönetilebilir hale gelecektir.

**6. Kullanıcı Arayüzü Tasarımı**

**6.1. Genel Yapı**

* **Layout yapısı**:
  + Üst menü (navbar): logo, kullanıcı menüsü, bildirim ikonu
  + Sol menü (sidebar): modül ve yetki bazlı menü öğeleri
  + Ana içerik alanı (content-body): seçilen sayfa dinamik olarak buraya yüklenir
* **Tasarım dili**:
  + Kurumsal tonlu (mavi/gri renk skalası)
  + Responsive (mobil uyumlu)
  + Role-based görünüm desteği

**6.2. Sayfa Akışı ve Wireframe Planı**

**Giriş ve Kayıt**

| **Sayfa** | **Açıklama** |
| --- | --- |
| Login.cshtml | E-posta ve şifre ile giriş ekranı |
| Register.cshtml | Yeni kullanıcı kaydı (firma oluşturma veya firmaya katılma seçimi) |
| JoinCompany.cshtml | Firma ID ile var olan bir firmaya katılma ekranı |
| ForgotPassword.cshtml | Şifre sıfırlama ekranı |
| ResetPassword.cshtml | Token ile şifre yenileme ekranı |

**Ortak Paneller**

| **Sayfa** | **Açıklama** |
| --- | --- |
| Dashboard.cshtml | Rol bazlı özelleşen ana panel: hızlı erişim kutuları, bildirimler, özet bilgiler |
| Notifications.cshtml | Gelen bildirimlerin listelendiği sayfa |
| Profile.cshtml | Kullanıcının kendi bilgilerini ve şifresini yönetebileceği alan |

**6.3. Firma Yönetim Ekranları (Admin)**

| **Sayfa** | **Açıklama** |
| --- | --- |
| CompanySettings.cshtml | Firma bilgileri güncelleme |
| CompanyModules.cshtml | Aktif modül yönetimi |
| UserManagement.cshtml | Kullanıcı daveti, rol atama, kullanıcı listesi |
| AuditLogs.cshtml | Sistem hareket geçmişi görüntüleme |

**6.4. İK Modül Ekranları**

| **Sayfa** | **Açıklama** |
| --- | --- |
| Employees/Index.cshtml | Çalışan listesi |
| Employees/Create.cshtml | Yeni personel ekleme |
| Departments/Index.cshtml | Departman listesi ve oluşturma |
| Positions/Index.cshtml | Pozisyon tanımlama ve güncelleme |
| Leaves/Index.cshtml | İzin talepleri listesi ve onay/reddetme |
| Leaves/MyRequests.cshtml | Çalışanın kendi izin geçmişi |
| Advances/Index.cshtml | Avans talepleri yönetimi |
| Payrolls/Index.cshtml | Bordro listesi ve detay görüntüleme |
| Performance/Index.cshtml | Performans değerlendirme giriş ve geçmişi |
| Reports/Index.cshtml | Raporlama ve istatistik ekranı |

**6.5. Genişletilmiş Modül Ekranları**

| **Sayfa** | **Açıklama** |
| --- | --- |
| Shifts/Index.cshtml | Vardiya tanımları |
| EmployeeShifts/Assign.cshtml | Çalışanlara vardiya atama |
| Attendance/Index.cshtml | Puantaj kayıtlarının görüntülenmesi |
| Recruitment/Posts.cshtml | İş ilanı oluşturma ve listeleme |
| Recruitment/Applications.cshtml | Başvuruların yönetimi |
| Documents/Index.cshtml | Evrak yükleme ve listeleme |
| Documents/MyDocuments.cshtml | Çalışanın kendi evraklarını görmesi |

**6.6. Rol Bazlı Arayüz Davranışı**

| **Rol** | **Girişte Göreceği Ana Ekran** |
| --- | --- |
| Firma Sahibi | Firma özeti, modül yönetimi, kullanıcı kontrolü |
| İK Yöneticisi | Personel listesi, izin/maaş/performans yönetimi |
| Müdür | Kendi departmanına ait işlemler, onay ekranları |
| Çalışan | Kendi izinleri, maaş bordrosu, evraklar, başvurular |

**6.7. UI Bileşen Notları**

* Tüm tablolar DataTables.js ile dinamik yapılacak (sıralama, filtreleme, sayfalama)
* Modal yapısı kullanılacak (örneğin: çalışan detayı, izin onayı)
* Role-based menu rendering yapılacak
* Yetki dışı sayfalar için erişim engeli mesajı gösterilecek

**7. API Entegrasyonları**

**7.1. Kullanılacak API'ler ve Amaçları**

| **API** | **Kullanım Alanı** | **Amaç** |
| --- | --- | --- |
| **MERNIS (Kimlik Doğrulama API)** | Personel kayıtları | TC Kimlik numarası üzerinden ad-soyad, doğum tarihi doğrulaması |
| **GİB (Vergi Kimlik Doğrulama API)** | Firma oluşturma | Vergi numarası ve unvan eşleşmesinin doğrulanması |
| **SGK İşe Giriş/Çıkış Bildirimi (planlanabilir)** | Personel bordrosu ve iş takibi | Personel işe giriş ve çıkışlarının otomatik olarak bildirilmesi |
| **MailKit (SMTP ile)** | Bildirim, şifre sıfırlama | Kayıt, onay, talep ve bildirim e-postalarının gönderilmesi |
| **SignalR** | Gerçek zamanlı bildirimler | Kullanıcılara anlık sistem mesajlarının gösterimi (örn. "İzin talebiniz onaylandı") |
| **PDFSharp (veya benzeri)** | Bordro çıktısı, rapor üretimi | Sistem üzerinden bordro ve performans formlarının PDF olarak oluşturulması |

**7.2. Entegrasyon Mimarisi**

* API çağrıları merkezi bir ApiService sınıfı üzerinden yapılacaktır.
* Dış API bağlantıları ayrı bir servis katmanında soyutlanarak, test edilebilirlik artırılacaktır.
* Gerekli durumlarda HttpClientFactory ile bağlantılar yönetilecek, yeniden kullanılabilir yapı sağlanacaktır.
* API bağlantı bilgileri Web.config dosyasında merkezi olarak tanımlanacak, güvenlik amacıyla şifreli saklanacaktır.

**7.3. Güvenlik ve Hata Yönetimi**

* API anahtarları (API Key, Bearer Token) ortam değişkeni veya güvenli config katmanında saklanacaktır.
* Her API entegrasyonu için hata günlüğü (ApiErrorLogs) tutulacaktır.
* API çağrılarında rate limit ve timeout yapılandırmaları uygulanacaktır.
* Kimlik doğrulama gereken API'ler için otomatik yenileme mekanizmaları (token refresh) uygulanabilir.

**7.4. Entegrasyon Geliştirme ve Test Süreci**

* Geliştirme sürecinde kullanılacak tüm dış servisler için test ortamları tercih edilecektir.
* Her entegrasyon mock veri ile test edilip sonra canlı bağlantıya geçirilecektir.
* API yanıtlarının şeması (JSON/XML) önceden dökümante edilerek model sınıfları oluşturulacaktır.

**8. Güvenlik (Yetkilendirme, Kimlik Doğrulama, SSL, vs.)**

**8.1. Kimlik Doğrulama (Authentication)**

* **Email + Şifre tabanlı kimlik doğrulama** uygulanacaktır.
* Kayıt olan her kullanıcı e-posta adresine gelen onay bağlantısını tıklayarak hesabını aktive edecektir.
* Girişlerde yanlış deneme sınırı olacak, belirli sayıda başarısız girişten sonra geçici engelleme uygulanacaktır.
* **Şifre sıfırlama** işlemleri için token tabanlı geçici bağlantı sistemi kullanılacaktır.
* Şifreler **salt + hash** yöntemiyle veritabanında saklanacaktır (BCrypt veya PBKDF2 önerilir).

**8.2. Yetkilendirme (Authorization)**

* Role-based yetkilendirme (RBAC) uygulanacaktır: Admin, İK Yöneticisi, Müdür, Çalışan
* Yetkilendirme hem arayüz düzeyinde (UI bazlı kontrol) hem de backend (controller bazlı kontrol) seviyesinde yapılacaktır.
* Role erişimi için ASP.NET MVC'deki [Authorize(Roles="...")] attribute’ü kullanılacak, dinamik erişimler için özel IAccessControlService sınıfı geliştirilecektir.
* Firma bazlı izinsiz erişimler için her işlemde CompanyId doğrulaması yapılacaktır.

**8.3. Erişim Kontrolleri**

* Her kullanıcı yalnızca bağlı olduğu firmaya ait verilere erişebilir.
* Yetkisi olmayan modüller sidebar ve dashboard’da görüntülenmeyecektir.
* API çağrılarında da modül ve rol denetimleri yapılacaktır.
* Kullanıcı girişlerinde IP loglama yapılacak, gerekirse coğrafi kısıtlama uygulanabilir.

**8.4. SSL / HTTPS Zorunluluğu**

* Uygulama, hem geliştirme hem de üretim ortamında **SSL (HTTPS)** üzerinden çalışacak şekilde yapılandırılacaktır.
* Tüm dış bağlantılar HTTPS üzerinden gerçekleştirilecektir.
* Tarayıcıda mixed content hataları engellenecek, tüm kaynaklar güvenli protokolden yüklenecektir.

**8.5. XSS / CSRF / SQL Injection Koruması**

* Tüm kullanıcı girişleri sunucu tarafında sanitize edilecektir.
* Razor View Engine ile oluşturulan formlarda CSRF token otomatik olarak kullanılacaktır.
* ORM olarak Entity Framework kullanıldığı için parametrik sorgular ile SQL injection riski minimize edilecektir.

**8.6. Parola Politikası**

* Minimum 8 karakterli karmaşık şifre zorunluluğu (büyük harf, küçük harf, rakam ve özel karakter)
* Şifreler kullanıcı arayüzünde asla açık gösterilmeyecek, resetleme dışında geri döndürülemeyecektir.
* Şifre sıfırlama linkleri yalnızca bir kez kullanılabilir ve süreyle sınırlıdır (örneğin 30 dakika geçerli).

**8.7. Loglama ve İzleme**

* Kullanıcı oturum bilgileri, erişim tarih/saat kayıtları AuditLogs tablosunda tutulacaktır.
* Kritik hatalar ve yetkisiz erişim denemeleri ayrıca günlük dosyasına (log file) ve/veya e-posta ile yöneticilere bildirilebilir.
* Gerektiğinde üçüncü taraf loglama sistemleriyle (örneğin ELK, Serilog) entegre edilebilecek yapı bırakılacaktır.

**9. Teknik Mimarisi ve Katmanlı Yapı**

ikCore, ASP.NET MVC (.NET Framework 4.8) üzerinde geliştirilecek, **N-Katmanlı Mimari** prensipleriyle yapılandırılacak ve SOLID prensiplerine uygun şekilde modüler olarak organize edilecektir. Uygulama, servis tabanlı bir yapı ile geliştirilecek ve ileride API tabanlı yapılara kolay geçişi destekleyecektir.

**9.1. Katman Yapısı**

Solution: ikCore

│

├── UI (ikCore.Web) → MVC arayüz (Views, Controllers)

├── Business → Servisler (iş mantığı)

├── DataAccess → Repository ve EF context yönetimi

├── Entities → Veri modelleri ve DTO’lar

├── Core (isteğe bağlı) → Ortak yardımcı sınıflar, Exception Handler, Logging vb.

**1. UI (Presentation Layer)**

* ASP.NET MVC kullanılarak geliştirilecek
* Razor View Engine ile oluşturulmuş sayfalar
* Bootstrap tabanlı responsive arayüz
* Controller’lar servis katmanı ile iletişim kurar (Business → Service Injection)

**2. Business (Service Layer)**

* İş kuralları bu katmanda uygulanır
* Her bir ana modül için ayrı servis sınıfları (EmployeeService, LeaveService, PayrollService vb.)
* Dependency Injection prensibi ile DataAccess katmanına bağımlıdır
* IService ve Service olarak interface-tabancı yapı kullanılır

**3. DataAccess (Repository Layer)**

* Entity Framework kullanılarak veritabanı işlemleri yürütülür
* Her tablo için ayrı repository sınıfı tanımlanır
* IRepository<T> → Generic Repository Pattern uygulanır
* Unit of Work deseni ile transaction yönetimi desteklenir

**4. Entities (Model Layer)**

* Tüm veri modelleri (User, Employee, LeaveRequest, vb.)
* ViewModel ve DTO sınıfları burada yer alır
* Fluent API veya Data Annotations ile veri kısıtlamaları tanımlanabilir

**5. Core (Yardımcı Katman – Opsiyonel)**

* Exception handling, custom logger, JWT utilities, constants, validators gibi tüm projelerde ortak kullanılacak yapılar
* Geliştirmenin ilerleyen safhalarında dahil edilebilir

**9.2. Kullanılacak Tasarım Desenleri**

| **Desen** | **Amaç** |
| --- | --- |
| **Repository Pattern** | Veri erişim işlemlerini soyutlamak |
| **Unit of Work** | Transaction yapısını bir bütün olarak yönetmek |
| **Dependency Injection** | Katmanlar arası bağımlılığı azaltmak, test edilebilirlik |
| **Factory Pattern** (isteğe bağlı) | Servis veya nesne örneklemelerini kontrol altında tutmak |
| **DTO / ViewModel Separation** | Arayüzde gösterilen verileri domain modellerinden ayırmak |

**9.3. Diğer Teknik Detaylar**

* Veritabanı işlemleri Entity Framework (Code First) yaklaşımıyla geliştirilecektir
* Arayüzde form validasyonları hem istemci (JavaScript/JQuery) hem de sunucu tarafında yapılacaktır
* Mail gönderimi MailKit veya System.Net.Mail üzerinden yapılacaktır
* Gerçek zamanlı işlemler için SignalR kullanımı planlanmaktadır
* Kullanıcı, firma ve modül bazlı erişim kontrolleri merkezi bir AccessControlService aracılığıyla sağlanacaktır

**10. Kullanılacak Teknolojiler**

**10.1. Yazılım ve Altyapı Teknolojileri**

| **Katman / Alan** | **Kullanılacak Teknoloji** |
| --- | --- |
| **Backend (Sunucu)** | ASP.NET MVC (.NET Framework 4.8) |
| **Frontend (Arayüz)** | Razor View Engine, HTML5, CSS3, JavaScript, jQuery, Bootstrap 5 |
| **ORM / Veritabanı Erişimi** | Entity Framework (Code First) |
| **Veritabanı** | Microsoft SQL Server |
| **Katmanlı Mimari** | N-tier Architecture (UI, Business, DataAccess, Entities) |
| **Bağımlılık Yönetimi** | Dependency Injection (Custom IoC veya Unity/Autofac) |
| **Repository Yapısı** | Generic Repository Pattern + Unit of Work |
| **Veri Doğrulama** | Data Annotations, Fluent Validation (isteğe bağlı) |

**10.2. Güvenlik ve Erişim Kontrolü**

| **Alan** | **Kullanılacak Teknoloji** |
| --- | --- |
| **Kimlik Doğrulama** | Forms Authentication, Cookie tabanlı oturum |
| **Yetkilendirme** | Role-Based Access Control (RBAC), [Authorize] attribute |
| **Şifreleme** | BCrypt veya PBKDF2 (şifre hashleme) |
| **SSL / HTTPS** | Sunucu yapılandırması üzerinden zorunlu SSL |
| **CSRF Koruması** | ASP.NET MVC Anti-Forgery Token |
| **Loglama** | Custom File Logging / Event Viewer / Serilog (isteğe bağlı) |

**10.3. Bildirim ve Gerçek Zamanlı İletişim**

| **Alan** | **Teknoloji** |
| --- | --- |
| **E-posta Gönderimi** | SMTP (System.Net.Mail veya MailKit) |
| **Gerçek Zamanlı Bildirim** | SignalR |
| **Şifre Sıfırlama** | Token tabanlı e-posta bağlantısı |

**10.4. API ve Dış Servis Entegrasyonları**

| **Entegrasyon Türü** | **Teknoloji / Servis** |
| --- | --- |
| **Mernis Kimlik Doğrulama** | Test ortamı API (KPS WS SOAP/REST) |
| **Vergi No Sorgulama (GİB)** | REST API |
| **Veri Formatları** | JSON (standart API veri alışveriş formatı) |
| **HTTP Tabanlı Erişim** | HttpClient, HttpClientFactory |