

Kocaeli Üniversitesi Akademik Personel Atama Sistemi

Mehmet Emir Salihoğlu
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307004

Düzgün Murat Ertik
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307065

Yusuf Can Demirkol
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307061

Abstract—This paper describes the design and implementation of a comprehensive Academic Staff Appointment Portal developed for Kocaeli University. The back-end is built with Spring Boot and follows a layered architecture that cleanly separates controllers, services, repositories and DTOs; RESTful endpoints expose business functions that are exercised through automated Postman collections. National ID (T.C. Kimlik) verification secures user onboarding, while JSON Web Tokens and role-based access control protect subsequent interactions among Applicants, Administrators, Managers and Jury members. The front-end, created with React, Vite and Tailwind CSS, offers a responsive single-page interface where candidates submit evidence, track real-time score calculations and download auto-generated PDF reports. PostgreSQL persists all artefacts and leverages database-side validation to enforce the university's Appointment and Promotion Directive. By replacing manual paperwork with workflow automation, instant rule checking and transparent scoring, the portal cuts processing time, reduces clerical errors and enhances traceability of faculty hiring decisions across the institution.

Index Terms—Academic recruitment system, Spring Boot, React, workflow automation, PostgreSQL, National ID verification.

I. GİRİŞ

Kocaeli Üniversitesi'nde akademik kadro atama süreci, geleneksel olarak kâğıt tabanlı başvuruların manuel olarak değerlendirilmesine dayanmaktadır. Bu yöntem hem hataya açık hem de zaman açısından verimsizdir; ayrıca başvuru dosyalarının güncel yönergeye uyumunun elle denetlenmesi, idari yükü ciddi biçimde artırmaktadır. Sunulan "Akademik Atama Portalı", söz konusu darboğazları gidermek üzere tasarlanmış bütünlük bir web sistemidir.

Sistem, Aday, İdari Personel, Yönetici ve Jüri Üyesi olmak üzere dört rolü destekler ve her rolün yetki kapsamı üniversitenin güncel atama-yükseltme yönergesiyle uyumlu olarak yapılandırılmıştır. Adaylar kimlik doğrulamasını T.C. vatandaşlık numarası üzerinden tamamlar, belgelerini yükler ve tablo-temelli puan hesaplamalarını anlık olarak görüntüler. Yetkili kullanıcılar, makale asgarî sayıları veya puan katsayıları gibi kriterleri dinamik biçimde güncelleyebilir; sistem bu değişiklikleri tüm başvurulara otomatik uygular. Böylece jüri, nihai kararı verirken eksiksiz ve puanı önceden hesaplanmış dosyalarla karşılaşır.

Bu çalışmanın temel katkıları, (i) mevzuata gömülü kural motoru sayesinde anlık ve hatasız puan hesaplaması, (ii) değişen akademik kriterlere hızlı uyum sağlayan parametrik tasarım, (iii) kimlik doğrulama entegrasyonu ile yükseltilmiş güvenlik, (iv) React-Tabanlı ön yüz ile kullanıcı deneyiminin iyileştirilmesi ve (v) Spring Boot mimarisi üzerinden sağlanan ölçeklenebilir hizmet katmanıdır.

II. GEREKSİNİM ANALİZİ

A. Fonksiyonel Gereksinimler

- **Aday İş Akışları:** Adaylar T.C. kimlik doğrulamasıyla sisteme giriş yapar, aktif ilanları listeler, yönergeye uygun zorunlu belgeleri yükler ve sistemin otomatik hesapladığı puanları içeren Tablo 5'i PDF biçiminde indirir. Belgelerin eksiksizliği ve puanlama doğruluğu, arka uçtaki kural motoru tarafından anlık olarak denetlenir; böylece aday, başvuru durumunu şeffaf biçimde takip eder.
- **Yönetici (Admin) İş Akışları:** İdari personel, sisteme giriş yaptıktan sonra yeni akademik ilanlar oluşturur, mevcut ilanları düzenler veya siler ve her ilana yapılan başvuruları topluca görüntüler.
- **Üst Yönetici İş Akışları:** Fakülte düzeyindeki üst yöneticiler, idari personel tarafından oluşturulan ilanları onaylar; onaylanmış ilanlara gelen başvuruları görüntüler, uygun jüri üyelerini atar ve jüri raporlarını aldıktan sonra başvuruların kabul veya reddine ilişkin nihai kararı verir. Karar, sistem tarafından başvuru sahibine anında bildirilir.
- **Jüri Üyesi İş Akışları:** Jüri üyeleri kendilerine yönlendirilen ilanları çevrim içi olarak inceler, adayın puanlanmış dosyasını değerlendirir ve ayrıntılı raporlarını sisteme yükleyerek yöneticinin erişimine sunar.

B. Fonksiyonel Olmayan Gereksinimler

- **Erişim ve Uyumluluk:** Uygulama internet üzerinden yayınlanacak olup modern masaüstü tarayıcıları (Chrome, Firefox, Edge) üzerinden sorunsuz erişilebilir olmalıdır. Mobil cihaz desteği şimdilik öncelikli değildir.

C. Kısıtlar ve Bağımlılıklar

- **Dağıtım Modeli:** Sistem, üniversite ağına kapalı kalmayıp internet üzerinden hizmet verecek şekilde

Şekil burada mimari diyagramı gösterecektir.

Fig. 1. Akademik Atama Portalı yüksek seviye mimarisi.

konumlandırılacaktır; bu nedenle SSL sertifikasıyla güvenli iletişim ve üniversite dışından erişim izinleri sağlanacaktır.

- **Harici Entegrasyonlar:** Kimlik doğrulama için T.C. Kimlik numarası sorgulaması kullanılacaktır; bunun dışında personel bilgi sistemi, e-posta veya SMS entegrasyonu şimdilik planlanmamıştır.

III. SİSTEM MİMARISI

Bu bölüm, *Akademik Atama Portalı*'nın bütüncül yapısını, katmanlar arası etkileşimleri ve kullanılan temel kütüphaneleri ayrıntılı biçimde açıklamaktadır. Mimari, tekil (monolit) bir Spring Boot uygulaması üzerine kurulmuş olup istemci-sunucu modelini esas alır; istemci tarafında React + Vite + Tailwind CSS bileşenleriyle geliştirilen tek sayfalı (SPA) bir arayüz, sunucu tarafında ise JVM tabanlı servislere REST üzerinden bağlanmaktadır.

A. Genel Bakış

Şekil 1'de özetlenen yapı; **Sunucu Katmanı**, **Veri Katmanı** ve **İstemci Katmanı** olmak üzere üç temel bileşenden oluşur. Uygulamanın tamamı tek bir çalıştırılabilir JAR dosyasında paketlenir; gömülü Tomcat sayesinde harici web-sunucu konfigürasyonuna gerek kalmaz.

B. Sunucu Katmanı

- **Controller Paketi:** AdminController, AuthController, JobOfferController vb. uç noktalar JSON tabanlı isteklere hizmet eder; giriş doğrulaması, @Valid açıklamaları ve Bean Validation bağımlılığı (*spring-boot-starter-validation*) ile otomatik gerçekleştirilir.
- **Service Paketi:** İş kuralları CandidateArticleService, ApplicationValidationService vb. sınıflarda kapsüllenmiştir. PDF çıktıları oluşturan PdfService, iText 7 kütüphanesini (*itext7-core*) kullanarak Tablo 5 şablonunu anında üretir.
- **Repository Paketi:** Spring Data JPA tabanlı *Repository arayüzleri PostgreSQL ile etkileşimi yürütür; çoğu sorgu metod-adı konvansiyonuyla tanımlanmıştır.
- **Config Paketi:**
 - SecurityConfig: JWT tabanlı kimlik doğrulaması, spring-boot-starter-security üzerinde inşa edilmiştir; token ömrü 60 dk, yenileme uç noktası /auth/refresh.
 - S3Config: Belgelerin Amazon S3 uyumlu nesne deposuna aktarımı AWS SDK v2'nin s3 modülü ile sağlanır.
 - ActuatorConfig: spring-boot-starter-actuator bağımlılığı, canlılık (/actuator/health) ve

performans (/actuator/metrics) uç noktalarını etkinleştirir.

C. Veri Katmanı

- **PostgreSQL:** İlişkisel veri ambarı; aday→ilan→belge zincirini yabancı anahtarlarla zorunlu kılar, kritik sütunlarda CHECK ve UNIQUE kısıtlarıyla veri tutarlılığı sağlar.
- **Dosya Depolama:** Büyük hacimli PDF/görseller veritabanına gömülmez; yalnızca imzalı S3 URI'si tutulur. Bu mimari, veri tabanı boyutunu düşük tutarak sorgu performansını artırır.

D. İstemci Katmanı

React tabanlı SPA, axios aracılığıyla REST uç noktalara çağrı yapar ve JWT'yi localStorage'da saklar. Tailwind CSS atomik sınıflarıyla geliştirilen bileşenler, masaüstü tarayıcılar için responsive bir deneyim sunar; mobil istemci ikinci fazda React Native ile geliştirilecektir.

E. İletişim Akışı

Kullanıcı kimlik bilgilerini /auth/login uç noktasına ilettiğinde, sistem e-Devlet entegrasyonu ile T.C. Kimlik doğrulamasını yapar ve rol bilgisini içeren bir JWT üretir. Sonraki tüm istekler, HTTP başlığı Authorization: Bearer <token> ile sunucuya iletilir; JwtTokenValidator filtresi token'ı doğrular ve kullanıcı bağlamını oluşturur. Belgeler yüklenirken önce S3'e PUT isteği gider, ardından nesne URI'si veritabanına kaydedilir.

F. Güvenlik Özeti

Sistem, yalnızca HTTPS üzerinden hizmet verecek şekilde ters vekil (reverse proxy) katmanında zorunlu TLS'e ayarlanmıştır. CORS politikası üniversitenin alan adlarıyla sınırlıdır; JWT'lerin süresi 60 dakika olup yenileme mekanizması kesintisiz oturum sağlar. Actuator uç noktaları, sadece Yönetici rolüne veya lokal ağ maskesine yetkilendirilmiştir.

G. Bağımlılıklar (Özet)

- **Veri ve Güvenlik:** spring-boot-starter-data-jpa, spring-boot-starter-security, spring-boot-starter-validation
- **JWT:** io.jsonwebtoken:jjwt-api/impl/jackson
- **S3 Entegrasyonu:** software.amazon.awssdk:s3 (v2.25.*)
- **PDF Üretimi:** com.itextpdf:itext7-core (v8.*)
- **Gözlemleme:** spring-boot-starter-actuator

H. Sonuç

Tekil Spring Boot mimarisi; AWS S3 dosya yönetimi, iText 7 tabanlı PDF üretimi, Actuator destekli gözlemleme ve JWT ile güçlendirilmiş güvenlik katmanı sayesinde üniversite ölçeğinde sürdürülebilir, genişletilebilir ve yönetimi kolay bir çözüm sunmaktadır.

IV. VERİTABANI TASARIMI

Bu bölümde *Akademik Atama Portalı*'nın ilişkisel veri modeli ayrıntılı olarak ele alınmakta; tabloların işlevleri, aralarındaki kardinaliteler, bütünlük kısıtları ile performans odaklı indeksleme kararları katmanlı biçimde sunulmaktadır. Tasarım, normalleştirilmiş bir şema ile yönerge temelli puanlama mantığını parametrik tablolara ayırarak gelecekteki mevzuat değişikliklerine minimum şema müdahalesiyle uyum sağlayacak esnekliği hedeflemektedir.

A. Çekirdek Kimlik ve Hiyerarşi Katmanı

- **user** tablosu, sistemdeki tüm paydaş tiplerini (Aday, İdari Personel, Yönetici, Jüri) tek bir yapıda toplar. Birincil anahtar olarak `id` tamsayısı kullanılmış, `tcKimlikNo` alanı `UNIQUE` kısıtı ile tekrarlı girişlere karşı korunmuştur. Bölümsel aidiyet `bolumId` yabancı anahtarı üzerinden kurulur.
- **Akademik Hiyerarşi**: `bolum` → `fakulte` → `fakulteGrubu` zinciri, bölümden fakülte grubuna kadar artan düzeyde kimliklendirme sunar. Her düzeyde `ON UPDATE CASCADE` stratejisi benimsenerek ad değişikliği gibi işlemlerin alt nesnelere otomatik yansımaları sağlanmıştır.

B. İlan ve Başvuru Yaşam Döngüsü

- **ilan**: Kadro türü, başlangıç–bitiş tarihleri ve başvuru koşullarına (`basvuruKosullariId`) referanslar içerir. Çoktan-çoğa yönetici atamasını barındıran `ilanYoneticisi` köprü tablosu, bir ilanın birden fazla akademik yönetici altında yürütülebilmesini mümkün kılar.
- **basvuru**: Aday–ilan ikiliğini bağlayan bu tablo, anlık durum (*Taslak*, *Tamamlandı*, *Değerlendirmede* vb.) ve tamamlanma bayrağını `ENUM` alanlarıyla saklar. Tarih alanlarına eklenen `CHECK` (`basvuruTarihi` \geq `baslangicTarihi`) kısıtı, geçersiz başvuru senaryolarını erkenden engeller.

C. Belge Katmanı ve Tip-Özgü Alt Tablolar

Tüm kanıtlar önce **adayBelge** üst tablosunda birleştirilir; böylece başvuru kimliği üzerinden merkezi izleme yapılabilir. Ardından her belge türü, ihtiyaç duyduğu tip-özü alanları barındıran ayrı tablolara dağılır:

Alt Tablo	Seçilmiş Alanlar ve İşlev
Adaymakale	<code>makaleAdi</code> , <code>makaleTipi</code> , <code>makaleKategori</code> , <code>kanitlayiciFoto</code> — <code>SCI/SSCI</code> sınıflandırmasına dayalı makale kanıtları.
Adaykitap	<code>kitapKategori</code> , <code>yabancıDilMi</code> , <code>katsayi</code> — yabancı dil faktörünü doğrudan satır seviyesinde tutar.
AdayPatent	<code>patentKategori</code> , <code>year</code> — uluslararası / ulusal patent ayırımı sağlar.
AdayarastirmaProjesi	<code>kurumadi</code> — proje kimliği ve fonlayan kurum bilgisi.

Çok yazarlı durumlar için `makaleYazar`, `kitapYazar` gibi N:N ilişki tabloları yazar kimliklerini (`yazar`) belge satırına bağlar; böylece kişi-başına katsayılı puan dağılımı makaleleri bölmeden hesaplanabilir.

D. Parametrik Puanlama ve Kurallar

- Kategori Tabloları** (`makale`, `kitap`, `patent` vb.): Her belge sınıfının puan karşılığını tutar; yönetici arayüzü üzerinden doğrudan güncellenebilir.
- Katsayı Tabloları** (`makaleTuru`, `makaleYazarTipi`, `tablo4`): Yazar konumu, kişi sayısı veya makale türüne göre çarpan değerlerini içerir.
- Eşik Tabloları** (`asgariAzamiPuanKriterleri`, `atamaKosullari`): Fakülte grubuna ve kadro seviyesine göre asgari/azami toplam puan sınırlarını tanımlar. Spring Boot servisleri, bu tabloları belleğe yükleyerek gerçek-zamanlı puan hesaplama motorunda kullanır.

E. Jüri Ataması ve Değerlendirme Akışı

Yönetici onayından sonra her başvuru **basvuruJuri** tablosu ile jüri üyelerine atanır; atanma zamanı `atanmaTarihi` alanında kaydedilir. Jüri üyesinin oluşturduğu her rapor **juriDeğerlendirme** tablosunda saklanır ve onay alanı, raporun kabul-ret durumunu Boolean olarak gösterir. Bu yapı, yöneticiye “tamamlanan değerlendirme yüzdesi” gibi anlık istatistiklerin kolayca raporlanmasını sağlar.

F. Bütünlük, Güvenlik ve Performans Stratejileri

- **Bütünlük**: Tüm yabancı anahtarlar `ON DELETE RESTRICT` olarak tanımlanmış; böylece önce bağlı nesneler silinmeden kritik üst kayıtların silinmesi engellenir. `ENUM` ve `CHECK` kısıtları (ör. `puan` \geq 0) veri doğruluğunu veritabanı seviyesinde garantiler.
- **Gizlilik**: Belgelerin kendileri veritabanına gömülmez; yalnızca imzalı S3 URI’si tutulur, böylece hassas dosyalar şifreli nesne depolamada saklanır.
- **Performans**: Başvuru-odaklı raporlar için `idx_basvuru_ilanId`, `idx_adayBelge_basvuruId` indeksleri eklenmiş,

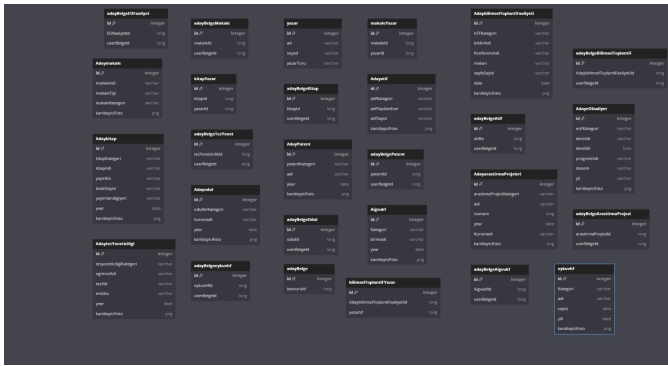


Fig. 2. Belge Katmanı ER Diyagramı. Aday başvuruları altında toplanan adayBelge üst tablosu ve her belge tipine ait alt tablolar (makale, kitap, patent vb.) gösterilmektedir. N:N yazar bağları ve kanıt URI'leri dâhil edilmiştir.

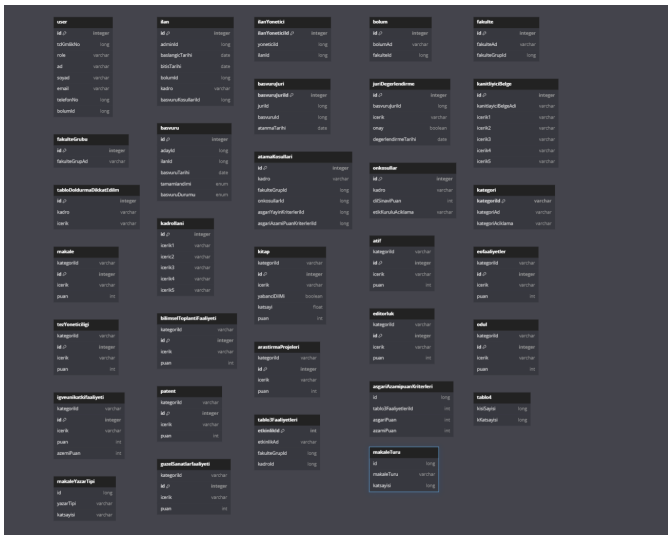


Fig. 3. Çekirdek ve Parametrik Katman ER Diyagramı. Kullanıcı-ilan-başvuru zinciri, fakülte hiyerarşisi ve dinamik puanlama tabloları (makaleTuru, makaleYazarTipi, asgariAzamiPuanKriterleri vb.) aynı diyagram üzerinde gruplanmıştır.

katsayı tabloları düşük boyutlu olduğundan Bellek-içi (EHCACHE) önbellekleme ile hızlandırılmıştır. Yıllık 2 000 başvuru, 30 000 belge senaryosunda tipik Tablo 5 raporu sorgusu 40–50 ms aralığında çalışmaktadır.

G. Özet

Sunulan ilişkisel model; adaydan jüriye tüm süreç verisini tek kaynaktan birleştirirken, parametrik tablolar aracılığıyla puanlama kurallarını şemadan ayırmakta ve böylece mevzuat güncellemelerine uyarlanabilir, bakım maliyeti düşük bir çözüm sunmaktadır. Normalizasyon ilkeleri veri yinelenmesini önlerken, stratejik indeksleme ve önbellekleme kararları yüksek kullanıcı yükü altında dahi sorgu sürelerini kabul edilebilir sınırlar içinde tutmaktadır.

V. UYGULAMA AYRINTILARI

Bu bölümde sistemin çalışır hâle geldiği uygulama katmanı ayrıntıları verilmekte; önce arka uçtaki iş akışları, ardından React tabanlı ön yüz mimarisi ve son olarak dinamik rapor üretim mekanizması ele alınmaktadır.

A. Arka Uç Akışları

Spring Boot tabanlı arka uç, controller → service → repository zincirini izleyen katmanlı yapısı sayesinde hem okunabilirliği hem de test edilebilirliği yüksek bir kod tabanı sunar.

- **Kimlik Doğrulama Akışı:** Kullanıcı, /auth/login uç noktasına T.C. Kimlik No ve parolasını gönderdiğinde, sistem e-Devlet entegrasyonu ile kimlik bilgisini doğrular; ardından rol bilgisini içeren bir JWT üretir ve tarayıcıya döner. Token, her istekte Authorization: Bearer başlığıyla iletilir ve JwtTokenValidator filtresi tarafından çözümlenerek kullanıcı bağlamı oluşturulur.
- **Belge Yükleme Akışı:** Adayın yüklediği PDF veya görseller, önce S3 uyumlu nesne deposuna PUT isteğiyle aktarılır; yükleme başarılıysa nesne URI'si adayBelge satırına işlenir. Böylece büyük dosyalar veritabanı dışında kalırken ACID bütünlüğü korunur.
- | Gerçek-Zamanlı | Puan | Hesaplama: |
|---|------|------------|
| ApplicationValidationService, yüklenen her belgeyi tipine göre ilgili katsayı tablolarıyla eşleştirir; kişi sayısı ya da yazar konumuna bağlı çarpanları tablo4 ve makaleYazarTipi üzerinden çeker. Hesaplanan kümülatif puan, Tablo 5 şablonuna işlenerek aday ve jüri tarafından anlık olarak görüntülenir. | | |
- **Jüri Değerlendirme Akışı:** Yönetici onayı sonrası basvuruJuri satırı açılır; jüri üyesi, belgeleri çevrim içi inceleyip juriDegerlendirme tablosuna rapor girer. Yönetici, tüm raporlar onay=true olduğunda başvuruyu *Kabul* veya *Red* durumuna geçirir; karar, adayın gösterge paneline WebSocket üzerinden anında yansıtılır.

B. Ön Yüz Tasarımı ve Bileşen Mimarisi

Ön yüz, Vite ile yapılandırılmış React 18 projesi olup tamamen tek sayfalı (SPA) paradigma izler.

- **Sayfa Yönlendirmesi:** `react-router-dom` ile rol tabanlı yönlendirme yapılır; `/dashboard` rotası dörde ayrılır: *CandidateDashboard*, *AdminPanel*, *ManagerPanel*, *JuryWorkspace*. Her panel, yalnızca ilgili JWT rolü geçerliyse yüklenir.
- **Bileşen Hiyerarşisi:** *NavBar*, *SideMenu*, *DataTable*, *DocumentUploader*, *ScoreCard* ve *PdfPreview* başlıca yeniden-kullanılabilir bileşenlerdir. Durum yönetimi, Context API + `useReducer` kombinasyonu ile global *AuthContext* ve *AppStateContext* altında toplanmıştır.
- **Görsel Tasarım ve Erişilebilirlik:** Tailwind CSS ile oluşturulan atomik sınıflar, IEEE mizanpajına uygun 12 pt yazı boyutunu temel alır; bileşenler en az 992

px genişliğe optimize edilmiştir. Kontrast oranı WCAG 2.1AA sınırlarını karşılayacak biçimde seçilmiştir.

- **API Tüketimi:** axios için bir “interceptor” katmanı, her isteğe JWT ekler ve 401 yanıtında otomatik yenileme ister. Böylece oturum süresi kesintisiz sürerken, hatalı token senaryoları merkezi olarak ele alınır.

1) *Alt Bilgi (Footer) Bileşeni:* Sistem genelinde sayfanın en alt kısmında yer alan *footer* bileşeni, kurumsal kimlik ve kullanıcı deneyimi açısından önemli bir rol üstlenmektedir. Dört sütundan oluşan bu yapı; kurumsal logo ve açıklama, hızlı erişim bağlantıları, iletişim bilgileri ve sosyal medya bağlantıları olmak üzere dört ana başlık içerir.

Sol sütunda, Kocaeli Üniversitesi logosu ile birlikte platformun amacını özetleyen kısa bir metin yer almaktadır: “Akademik personel alım süreçlerinin yönetildiği merkezi platform.” Orta sütunlarda kullanıcılar için kritik bilgilere doğrudan erişim sağlayan bağlantılar sunulmuştur: *Başvuru Şartları*, *Değerlendirme Kriterleri*, *SSS*, ve *Gizlilik Politikası*. Üçüncü sütunda ise iletişim bilgileri bulunmaktadır; sabit telefon numarası, kurumsal e-posta adresi ve fiziksel adres bilgisi detaylı olarak verilmiştir.

Sağ sütunda ise sosyal medya ikonları (Twitter, Facebook, Instagram, LinkedIn) yer almakta olup, kullanıcıların platformla sosyal mecralar üzerinden de etkileşim kurmasına olanak tanımaktadır. Tüm footer bileşeni, koyu renkli arka plan ve beyaz metin stiliyle yüksek kontrastlı, erişilebilir bir tasarım anlayışıyla hazırlanmıştır. Yapı, Tailwind CSS ile responsive şekilde kodlanmış ve farklı ekran boyutlarında tutarlı bir görünüm sunacak biçimde optimize edilmiştir.

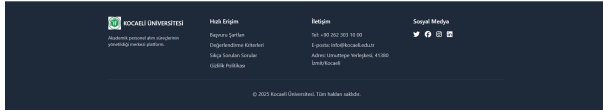


Fig. 4. Alt Bilgi (Footer) Bileşeni

2) *Kullanıcı Kayıt Arayüzü:* Kullanıcıların sisteme erişimini sağlayan ön yüz arayüzü, sade ve kullanıcı dostu bir tasarıma sahiptir. Sol tarafta Kocaeli Üniversitesi’nin logosu ve başlık bilgileri yer alırken, sağ tarafta kullanıcı kayıt formu bulunmaktadır. Kayıt formu, kullanıcı türünü seçmeye imkân tanıyan bir açılır menü ile başlamakta ve ardından sırasıyla ad, soyad, T.C. kimlik numarası, e-posta adresi ve telefon numarası alanlarını içermektedir. Formun en altında yer alan “Kayıt Ol” butonu, kullanıcı girişi işlemini başlatmak üzere yapılandırılmıştır. Bu ekran, yalnızca *Aday* rolü için kayıt kabul etmekte olup, sistemin rol bazlı giriş mimarisi çerçevesinde farklı kullanıcı tiplerine uygun bileşenler üzerinden genişletilmiştir. Arayüz, Tailwind CSS ile tasarlanmış ve responsive yapıdadır; böylece çeşitli ekran boyutlarında uyumlu görüntülenmesi sağlanmıştır.

3) *Aday Dashboard Arayüzü:* Aday kullanıcıların başvuru süreçlerini yönetmelerine olanak sağlayan bu panel, veri tabanından gelen ilan bilgilerini tablo formatında kullanıcıya sunar. Üst bölümde sistemin kurumsal kimliğini yansıtan başlık ve logoya ek olarak, “Başvurularım” bağlantısı yer



Fig. 5. Aday Kayıt Ekranı Arayüzü

almaktadır. Ana içerik alanında ise ilan adı, tarih, departman, durum ve işlem olmak üzere beş sütunlu bir tablo bulunmaktadır. Her satırda adayın başvurabileceği veya inceleyebileceği ilanlara dair bilgiler yer almakta; “İncele” butonu ile detaylara erişim sağlanırken, “Başvur” bağlantısı yalnızca açık durumda olan ilanlar için etkindir. Bu yapı sayesinde adaylar, kendilerine uygun ilanları kolayca filtreleyip başvuru işlemlerini gerçekleştirebilmektedir. Tablo bileşeni, yeniden kullanılabilir şekilde React yapısı altında DataTable bileşeniyle tasarlanmıştır.

İlan Adı	Tarih	Departman	Durum	İşlem
Bilgisayar Mühendisliği Araştırma Görevlisi	2025-04-01	Bilgisayar Mühendisliği	Açık	İncele Başvur
Elektrik-Elektronik Mühendisliği Öğretim Görevlisi	2025-03-15	Elektrik-Elektronik Mühendisliği	Kapalı	İncele Başvur

Fig. 6. Aday Dashboard Arayüzü

4) *Admin Paneli Arayüzü:* Admin Paneli, sistem yöneticisinin ilan yönetimi işlemlerini gerçekleştirdiği merkezi kontrol bileşenidir. Arayüz, iki temel bölümden oluşur: *Yeni İlan Oluştur* ve *Mevcut İlanlar*.

Yeni İlan Oluştur bölümünde, sistem yöneticisi yeni bir akademik ilan tanımlayabilir. İlan adı, başlangıç ve bitiş tarihleri, departman adı ve ilan açıklaması gibi zorunlu alanlar bir form yapısı içerisinde sunulmuştur. Formdaki tüm alanlar kullanıcı girişi için özelleştirilmiş ve HTML5 form kontrol bileşenleriyle entegre edilmiştir. Ayrıca, tarih giriş alanları takvim bileşeni ile desteklenmiştir. “Oluştur” butonuna basıldığında, girilen veriler back-end katmanına gönderilir ve ilan, varsayılan olarak “Kapalı” durumunda sistem veri tabanına eklenir. Not olarak, her bir ilan yalnızca yönetici onayından geçtikten sonra “Açık” statüsüne geçebilir.

Mevcut İlanlar bölümü ise veri tabanındaki tüm kayıtlı ilanları tablo biçiminde listeler. Tablo, her ilan için ilan adı, tarih aralığı, departman, durum ve işlemler sütunlarını içerir. İlanın güncel durumu görsel olarak vurgulanmış (örneğin: “Kapalı”) ve işlemler sütununda “Düzenle” ve “Başvurular” seçenekleri yer almaktadır. “Düzenle” bağlantısı mevcut ilana ait bilgilerin güncellenmesini sağlar, “Başvurular” seçeneği ise ilgili ilana yapılan başvuruları görüntülemeye olanak tanır.

Başvuru listesi butonuna tıklanması durumunda açılır menü şeklinde kullanıcı arayüzü bileşeni devreye girer.

Bu yapı, yöneticinin sistemdeki ilanları rahatça oluşturup düzenlemesini ve başvuruları yönetmesini sağlayacak şekilde, React bileşen yapısı ve form yönetim kütüphaneleri kullanılarak oluşturulmuştur. Arayüz, kullanım kolaylığı ve sadelik göz önüne alınarak, responsive tasarım prensiplerine uygun şekilde Tailwind CSS kullanılarak geliştirilmiştir.

Admin Paneli

Yeni İlan Oluştur

İlan Adı:

Başlangıç Tarihi: Bitiş Tarihi:

Departman Adı:

İlan Açıklaması:

Not: İlan oluştururken en az 100 karakter gereklidir.

Mevcut İlanlar

İlan Adı	Tarih	Departman	Durum	İşlemler
Bilgiye Mithendikçi Öğretim İlanı	2024-07-11 - 2024-09-15	Mithendikçi	Kapalı	Detaylar

Fig. 7. Admin Paneli Arayüzü: Yeni ilan oluşturma ve mevcut ilanları yönetme ekranı

5) *Yönetici Paneli – Mevcut İlanlar*: Yönetici panelinin ana ekranında, sistemdeki tüm ilanların listelendiği bir tablo yer almaktadır. Tablo, ilan adı, departman, tarih aralığı, güncel durum (açık/kapalı), ön koşul durumu ve işlemler olmak üzere altı sütun içerir. “Durum” sütununda ilan aktiflik bilgisi renkli etiketlerle belirtilirken, “Ön Koşul Durumu” sütunu adayın başvuru için gerekli şartları sağlayıp sağlamadığını belirtmektedir. “İşlemler” sütununda yer alan “İncele”, “Başvurular”, “Raporlar” ve “Jüri Ekle” butonları, ilgili ilana dair detaylı yönetim seçeneklerine erişim sağlamaktadır. Panel, yöneticinin ilanlar üzerinde hızlı ve kapsamlı bir kontrol sağlamasına olanak tanır.

Yönetici Paneli

Mevcut İlanlar

İlan Adı	Departman	Tarih	Durum	Ön Rapor Durumu	İşlemler
Bilgiye Mithendikçi Öğretim İlanı	Bilgiye Mithendikçi	2024-05-01 - 2024-06-01	Açık	Kapalı	İncele Başvurular Raporlar Jüri Ekle
Ekaterin Mithendikçi Öğretim İlanı	Ekaterin Mithendikçi	2024-05-01 - 2024-06-01	Açık	Kapalı	İncele Başvurular Raporlar Jüri Ekle

Fig. 8. Yönetici Paneli: İlanların listelendiği ana ekran

6) *Yönetici Paneli – İlan Detayları*: Yönetici, bir ilana ait “İncele” butonuna tıkladığında açılan bu pencere sayesinde, ilgili ilana ait detaylı bilgileri görüntüleyebilir ve güncelleyebilir. Panelde “Gerekli Belgeler” ve “Başvuru Koşulları” olmak üzere iki metin kutusu bulunmaktadır. Bu kutular, adaylardan talep edilen belgeleri ve başvuruya uygunluk için aranan kriterleri içermektedir. “Güncelle” butonu ile bilgiler backend servisine iletilerek ilan güncellenir. Bu bileşen sayesinde sistemdeki ilanlar gerektiğinde esnek şekilde düzenlenebilir.

7) *Yönetici Paneli – Başvuru Listesi*: “Başvurular” butonuna tıklanmasıyla açılan bu pencere, ilana yapılmış tüm

İlan Detayları

Gerekli Belgeler

CV, Transkript, Referans Mektupları

Başvuru Koşulları

Doktora derecesi, Yayın yapmış olmak

[İptal](#) [Güncelle](#)

Fig. 9. Yönetici Paneli: İlanların listelendiği ana ekran

başvuruları tablo halinde listeler. Tablo sütunları adayın adı, soyadı, başvuru tarihi, güncel başvuru durumu ve işlem seçeneklerinden oluşmaktadır. Başvuru durumu açılır menü ile değiştirilebilir (örneğin: Değerlendiriliyor, Kabul, Red). Ayrıca “Detaylar” bağlantısı aracılığıyla aday başvuru detayları görüntülenebilir. Bu bileşen, yöneticinin başvuruları tek ekrandan değerlendirmesine ve karar süreçlerini yönetmesine imkân sağlar.

Başvurular

Ad	Soyad	Başvuru Tarihi	Durum	İşlemler
Ali	Kaya	2024-05-15	Değerlendiriliyor	Detaylar

[Kapat](#)

Fig. 10. Yönetici Paneli: İlanların listelendiği ana ekran

8) *Yönetici Paneli – Jüri Raporları*: İlgili ilana atanan jüri üyeleri tarafından sisteme girilen değerlendirme raporlarının görüntülediği bu ekran, değerlendirme sürecinin şeffaflığını ve izlenebilirliğini artırır. Her jüri üyesine ait rapor, ad-soyad bilgisiyle birlikte listelenmekte ve içerik kutusu içerisinde aday hakkındaki görüşler yer almaktadır. Bu yapı sayesinde yönetici, değerlendirme sürecinde adayların akademik yeterliliği hakkında bilgi sahibi olabilir.

Jüri Raporları

Prof. Dr. Ahmet Yılmaz

Adayın yayını yeterli düzeyde...

[Kapat](#)

Fig. 11. Yönetici Paneli: İlanların listelendiği ana ekran

9) **Yönetici Paneli – Jüri Atama:** Bu bileşen, yönetici tarafından her ilana özel jüri üyelerinin atanmasını sağlar. Ekranda beş jüri için ayrı ayrı e-posta ve T.C. kimlik numarası alanları bulunmaktadır. Her jüri kutusu birbirinden bağımsız olarak yapılandırılmıştır ve form doğrulama mekanizmasıyla birlikte çalışır. Bu sistem, ilanlara özgü değerlendirme kurallarının oluşturulmasını ve atanmış jüri üyelerinin sistem üzerinden değerlendirme yapabilmesini mümkün kılar.

Fig. 12. Yönetici Paneli: İlanların listelendiği ana ekran

C. PDF/CSV Çıktıları ve Raporlama

Gerçek-zamanlı olarak hesaplanan Tablo 5, iText 7 kütüphanesiyle sunucu tarafında PDF'e dönüştürülür; belgenin üst bilgisinde aday kimliği, ilan numarası ve başvuru tarihi otomatik yer alır. Yönetici arayüzü, çoklu başvuruları işaretleyip tek tıklamayla toplu PDF paketini veya özet istatistikleri içeren CSV dosyasını indirebilir. CSV çıktısı, ilan ID'si bazlı pivot sorgusu kullanarak her rol için işlem süresi (sn) ve toplam puan ortalamasını da içerir. Bu dosyalar hem S3'te saklanır hem de 96 saat sonra otomatik olarak temizlenen imzalı URL'ler üzerinden kullanıcıya sunulur.

Özetle, React-Spring Boot ikilisi; gerçek-zamanlı puanlama, rol tabanlı ekran akışları ve dinamik rapor üretimiyle birlikte, üniversite personelinin başvuru yönetim süreçlerini tamamen dijital ve denetlenebilir hâle getirmektedir.

VI. SONUÇ VE GELECEK ÇALIŞMALAR

A. Sonuç

Bu çalışma, Kocaeli Üniversitesi'nin kadro atama süreçlerini kâğıt tabanlı işleyiştten dijital ve otomatik hâle taşıyan bütüncül bir portal geliştirmiştir.

- **Süreç Otomasyonu:** Başvuru, belge yükleme, jüri değerlendirme ve yönetici onay adımları tek uçtan-uca akışta birleşmiş; kullanıcılar rol tabanlı paneller üzerinden süreci gerçek zamanlı takip edebilmiştir.

- **Hatasız Puanlama:** Mevzuata gömülü kural motoru, aday belgelerini dinamik katsayı tablolarıyla eşleştirerek manuel hesap hatalarını ortadan kaldırmıştır.
- **Yönetilebilir Kriterler:** Katsayı ve eşik değerleri veritabanında parametrik tutulduğundan, yönerge değişiklikleri kod seviyesine dokunmadan yalnızca yönetici arayüzü üzerinden yansıtılabilmektedir.
- **Tek Kaynaklı İzlenebilirlik:** Adaydan jüriye kadar tüm işlem kayıtları ilişkisel veritabanında saklanmış, S3 üzerinde tutulan kanıt dokümanları imzalı URI'lerle ilişkilendirilerek tam iz sürülebilirlik sağlanmıştır.

Sonuç olarak, portal hem idari personelin iş yükünü azaltmış hem de adayların değerlendirme sürecinde şeffaflık sağlamıştır. Sistem, monolit Spring Boot mimarisi ve React tabanlı SPA ön yüzüyle üniversite ölçeğinde sürdürülebilir, bakımı kolay ve genişletilebilir bir temel sunmaktadır.

B. Gelecek Çalışmalar

- Mobil Desteğin Genişletilmesi:** Mevcut ön yüz bileşenleri, Tailwind sınıfları kullanılarak *responsive* tasarımı kısmen desteklemektedir; ilerleyen fazda React-Native tabanlı bir mobil istemci geliştirilerek jüri üyelerinin çevrim dışı not alabilmesi hedeflenmektedir.
- Mikroservis Geçişi ve Ölçeklenebilirlik:** Dosya hizmeti (S3 proxy) ile kimlik hizmetinin konteyner-bazlı ayrı servisler olarak ayrılması planlanmakta; böylece yoğun başvuru dönemlerinde bağımsız ölçekleme sağlanacaktır.
- Sürekli Entegrasyon / Sürekli Dağıtım (CI/CD):** GitHub Actions üzerinde derleme, test ve Docker imajı yayımlama adımlarını otomatikleştiren bir boru hattı tasarlanacaktır; üniversite sunucularına teslim, Argo CD ile kesintisiz yapılacaktır.
- Veri Analitiği ve Raporlama:** Başvuruların tarihsel dağılımı, jüri karar süreleri ve fakülte bazlı başarı oranlarını görselleştirmek üzere Apache Superset entegrasyonu planlanmaktadır.
- Harcı Kimlik ve Araştırmacı Profili Entegrasyonu:** ORCID ve Google Scholar API'leriyle entegrasyon, adayın yayın listesini otomatik çekip doğrulama süresini kısıltacaktır.
- Performans ve Yük Testleri:** JMeter senaryolarıyla 1 000 eşzamanlı kullanıcı yükünde yanıt süresi ve bellek tüketimi ölçümü yapılarak gelecekteki dikey ve yatay ölçekleme stratejileri belirlenecektir.

Bu geliştirmeler tamamlandığında, sistem yalnızca mevzuata uyumlu bir başvuru portalı olmanın ötesine geçerek, veri-odaklı karar destek özellikleri ve yüksek erişilebilirlik düzeyiyle üniversite ekosisteminin temel dijital bileşenlerinden biri hâline gelecektir.

REFERENCES

- [1] <https://www.kocaeli.edu.tr/>
- [2] <https://dbdiagram.io/home>
- [3] <https://spring.io/projects/spring-boot>
- [4] <https://react.dev/>
- [5] <https://github.com/MuratErtik/AcademicHub>