

# Akademik Personel Başvuru ve Değerlendirme Sistemi

Yazılım Geliştirme Laboratuvarı I (Bahar)

Efe Tuna Günay  
Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği  
221307069

efetunagunayyy@gmail.com

Yunus Emre Kılıç  
Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği  
221307062

yunus13e0@gmail.com

## Abstract

This study presents a secure and role-based academic staff recruitment system where candidates can upload their CVs and apply for academic positions. Administrators can assign juries to job announcements, and jury members evaluate applicants by scoring and generating PDF reports. The system ensures identity validation using the Turkish Republic Identity Number SOAP service and separates access for applicants, juries, managers, and admins. Built using Next.js, TypeScript, Express.js, and Prisma ORM, the platform provides a modern and scalable approach to managing academic recruitment processes.

**Keywords** — academic recruitment, role-based access, CV upload, SOAP service, jury scoring, PDF reporting

## I. GİRİŞ

Üniversitemizde yürütülen akademik personel alım süreci, çoğu zaman manuel ve belgelerle ilerlemektedir. Bu proje, başvuru sürecini dijitalleştirerek adayların özgeçmiş yükleyebildiği, ilânlara başvurduğu ve başvuru durumlarını takip edebildiği bir sistem sunmaktadır. Jüriler kendilerine atanan başvuruları değerlendirip puan verir, PDF formatında rapor hazırlar ve sonucu iletir. Sistem, her kullanıcının rolüne göre yetkilendirilmiş olup, kimlik doğrulama işlemiyle resmi verilerin kontrolü sağlanmaktadır.

## II. YÖNTEM

Sistem üç ana kullanıcı tipini içermektedir:

- Aday
- Jüri Üyesi
- Yönetici ve Admin

### Bu sistemle:

- Adaylar başvuru sürecini dijital ortamda takip edebilir.
- Özgeçmiş yüklemesi ve ilanlara başvuru yapabilir.
- Jüri üyeleri sadece atandıkları ilanlara erişim sağlar.

- Jüri puanlama ve PDF rapor sunar.
- Yöneticiler ilanlara jüri atar, admin yeni ilan oluşturur.
- TC Kimlik doğrulama SOAP servisiyle gerçekleştirilir.
- Jwt ile oturum güvenliği sağlanmıştır.

## III. KULLANICI ROLLERİ VE İŞLEVLERİ

Rol	Yetkiler
Aday	CV yükleme, ilanlara başvuru, durum takibi
Jüri	Atandığı ilanlardaki başvuruları puanlama, PDF rapor
Yönetici	Jüri atama, ilan üzerinden denetim
Admin	Yeni ilan oluşturma

## IV. SİSTEM MIMARISI

Sistem, modern web teknolojileri ile ölçeklenebilir şekilde geliştirilmiştir:

- Frontend:** Kullanıcı arayüzü için Next.js, TypeScript, Tailwind CSS tercih edilmiştir. Bu yapı, modüler ve hızlı yüklenen bir deneyim sunar.
- Backend:** Sunucu tarafı işlemler Node.js üzerinde çalışan Express.js ile yazılmıştır. Kimlik doğrulama, veri işleme ve kullanıcı yönetimi bu katmanda gerçekleştirir.
- Veritabanı:** PostgreSQL veritabanı Prisma ORM ile entegre edilmiştir. Bu sayede veri modelleri TypeScript ile senkronize çalışmakta ve veri doğrulama kolaylaşmaktadır.
- Kimlik Doğrulama:** Türkiye Cumhuriyeti Kimlik No doğrulaması için Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü'nün sunduğu SOAP servisi kullanılmaktadır. Bu servis XML tabanlıdır ve

kullanıcının adı, soyadı ve doğum yılı ile birlikte çalışır.

- **PDF Üretimi:** Jüri üyeleri tarafından puanlanan başvurular, **pdf-lib** kütüphanesi aracılığıyla PDF formatında rapora dönüştürülür.
- **JWT (JSON Web Token):** Kullanıcıların rol bazı giriş işlemleri JWT üzerinden yapılmakta ve yetki denetimleri buna göre filtrelenebilmektedir.

#### A. Frontend

Frontend uygulaması, sistemde tanımlı kullanıcı rollerine göre (aday, jüri, yönetici, admin) farklı arayüzler sunmak üzere **Next.js** (App Router mimarisi)[1] ve **TypeScript** ile geliştirilmiştir. Kullanıcıların sistemle olan tüm etkileşimleri bu arayüzler üzerinden gerçekleştirilir. Arayüz bileşenleri modüler, yeniden kullanılabilir ve sürdürülebilir olacak şekilde yapılandırılmıştır. Tasarımda **TailwindCSS** ile modern, sade ve erişilebilir bir kullanıcı deneyimi sağlanır.

Uygulama açılışında kullanıcıdan kimlik bilgileri alınır. Kimlik doğrulaması başarılı olursa, kullanıcı rolüne göre sistemi kullanır. Önyüzde yazılanlar:

#### Üye Ol Ekranı:

Kullanıcı ad, soyad, TC kimlik numarası ve doğum tarihi ile devlet vatandaşlık doğrulamasından geçer. Şifre oluşturur ve sisteme kayıt olur.

#### Giriş Ekranı:

Daha önceden kayıt olan kullanıcı burada TC kimlik numarası ve şifresini girerek sisteme giriş yapar.

#### Aday Paneli:

Başvuru yapılabilecek ilanlar listelenir.

Gerekli çalışmaların, bilgilerin yükleme işlemi yapılabilir. Gerekli olan bütün faaliyetlerin bilgi girişi component olarak oluşturulup burada kullanılmıştır.

Daha önce yapılan başvurular ve durumları (örneğin: "Bekliyor", "Kabul Edildi", "Reddedildi") görüntülenir.

Başvuruların detayları incelenebilir.

#### Jüri Paneli:

Sadece atandığı ilanlar ve bu ilanlara yapılmış başvurular görünür.

Her başvuru için puanlama ekranı sunulur. Puanlama kategori bazı yapılır.

PDF rapor oluşturma butonu ile değerlendirme süreci tamamlanır.

#### Yönetici Paneli:

İlanlara jüri ataması yapılır.

#### Admin Paneli:

Yeni ilanlar oluşturulabilir.

#### B. Backend

Backend, tüm kullanıcı rolleri için veri yönetimini ve iş mantığını sağlayan merkezi API katmanıdır. Node.js üzerinde **Express.js**[2] framework'ü ile geliştirilmiş olup, veritabanı etkileşimi için **Prisma ORM** kullanılmaktadır.

Backend üç ana sorumluluğa sahiptir:

- Kimlik doğrulama ve yetkilendirme (JWT temelli)
- İş mantığı yönetimi (başvuru, puanlama, jüri atama vb.)
- Harici servis entegrasyonları (SOAP servisi, dosya yönetimi, PDF üretimi)

Kod yapısı modülerdir:

- **routes/:** Her rol için ayrı API route'ları (aday, jüri, admin)
- **controllers/:** İş mantığını içeren fonksiyonlar
- **services/:** Kimlik doğrulama, PDF üretimi, SOAP entegrasyonu
- **prisma/:** Veritabanı modelleri ve ilişkiler

#### Kimlik Doğrulama ve Yetkilendirme

- Kullanıcı girişi ile birlikte **bcrypt** ile hash'lenmiş şifre doğrulanır.
- Başarılı girişte, kullanıcıya rolü ve ID bilgilerini içeren bir **JWT token** verilir.[3]

#### Kimlik Doğrulama (SOAP Servisi)

- Kullanıcının T.C. Kimlik Numarası ve kimlik bilgileri, **tckimlik.nvi.gov.tr** adresindeki **SOAP XML servisi** üzerinden doğrulanır.[4]
- Backend, **xml2js** kütüphanesini kullanarak SOAP mesajlarını oluşturur ve sonucu parse eder.
- Hatalı giriş durumunda kullanıcıya açıklayıcı hata mesajı döner.

#### Başvuru ve Faaliyet Yükleme Süreci

- Aday, ilanlara başvuru yaparken gerekli faaliyet bilgilerini girer. Hepsini için tek tek service ve controller yazılmıştır.
- CV dosyasına sadece yetkili jüri ve yöneticiler erişebilir.

#### Jüri Atama ve Puanlama

- Yönetici, sistemdeki aktif ilanlara jüri ataması yapar.
- Jüri, yalnızca atandığı ilanlara başvuran adayları puanlayabilir.

- Puanlama belirli kategorilere göre yapılır ve her kategori ayrı bir satır olarak puanlama tablosuna yazılır.
- Puanlamalar tamamlandığında, jüri tarafından tek tuşla PDF raporu oluşturulur.

#### *Admin ve İlan Yönetimi*

- Admin, sistem üzerinden yeni ilanlar oluşturabilir. Her ilan; başlık, açıklama içerir.
- İlanlar aktif/pasif duruma getirilebilir.
- İlgili veriler Prisma ORM ile ilan tablosuna yazılır.

#### *Veri Tabanı İlişkileri ve Güvenliği*

- Tüm veritabanı işlemleri Prisma ORM ile gerçekleştirilmiştir. [5]
- Modeller arası ilişkiler açıkça tanımlanmıştır (örneğin: Başvuru ↔ Kullanıcı, Puanlama ↔ Jüri)

#### *Hata Yönetimi ve Loglama*

- Her API çağrısı özel bir try-catch bloğunda çalıştırılır.
- Hatalar kullanıcıya standart formatta ({success: false, message: "..."}) döndürülür.
- Sunucu taraflı hatalar loglanır ve geliştirici tarafından takip edilebilir hale getirilir.

### V. SİSTEM İŞ AKIŞI

1. Aday, sisteme giriş yapar ve kimlik doğrulaması yapılır.
2. Başarılı doğrulamanın ardından aday, sisteme CV yükler.
3. Aday, uygun gördüğü açık ilanlara başvuru yapar.
4. Yönetici, her ilan için uygun jüri üyelerini sistem üzerinden atar.
5. Jüri üyeleri, atandıkları başvurulara ait aday bilgilerini inceler ve belirlenen kriterlere göre puanlama yapar.
6. Yapılan değerlendirmeler sonucunda sistem otomatik olarak PDF formatında jüri raporları üretir.
7. Başvuruların durumu jüri değerlendirmesine göre güncellenir ve adaylar bilgilendirilir.

8. Admin kullanıcı, yeni ilanlar oluşturabilir ve sistem genelini denetler.

### VI. GÜVENLİK VE TEST

- **JWT ile Kimlik Denetimi:** Kullanıcı girişlerinde, oturum boyunca geçerli olacak imzalı JWT token'ları kullanılır. Bu token'lar aracılığıyla hem kimlik doğrulama hem de rol bazlı erişim kontrolü sağlanır.
- **bcrypt ile Şifre Hashleme:** Kullanıcı şifreleri, bcrypt algoritması ile güçlü şekilde hash'lenerek veritabanında saklanır. Bu sayede veri ihlallerinde şifrelerin açığa çıkması önlenir.
- **Rol Tabanlı Yetki Kontrolü:** Kullanıcıların sistemde hangi işlemleri yapabileceği rollerine göre kısıtlanır. Örneğin bir jüri üyesi, sadece atandığı başvuruları görebilir; aday kendi başvurularını takip edebilir.
- **Kimlik Doğrulama Testi:** TC Kimlik Numarası doğrulaması, T.C. İçişleri Bakanlığı'na ait SOAP web servisi üzerinden test edilmiş ve geçerli bilgilerle başarılı doğrulama sağlanmıştır.
- **JWT Doğrulama Testi:** JWT token üretimi ve çözümleme işlemleri test edilmiş, hatalı token ile erişim denemelerinde sistemin koruma sağladığı gözlemlenmiştir.
- **Puanlama Tutarlılığı:** Her jüri sadece kendi atandığı başvurulara erişebilmiştir.

### VII. SONUÇ

Bu proje, akademik personel başvuru süreçlerinin dijital ortama taşınarak şeffaf, hızlı ve güvenli şekilde yürütülmesini sağlamayı amaçlamıştır. Kimlik doğrulama giriş sistemi, rol bazlı yetkilendirme, jüri puanlama sistemi ve otomatik PDF raporlama gibi özellikler, sürecin dijitalleştirilmesinde önemli katkılar sunmaktadır. Hem adaylar hem de jüri üyeleri için kullanıcı dostu bir deneyim sağlarken, yöneticiler için de kontrol ve denetim kolaylaştırılmıştır.

#### Kaynakça

- [1] <https://nextjs.org/>
- [2] <https://expressjs.com/>
- [3] <https://jwt.io/>
- [4] <https://tckimlik.nvi.gov.tr/>
- [5] <https://www.prisma.io/>
- [6] <https://www.postman.com/>

Github linkleri: <https://github.com/efetuna1/akademik-atama-frontend>  
<https://github.com/efetuna1/akademik-backend>