

Akademik Personel Başvuru Portalı

Miray Gürbüz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307031

Zeynep Yılmaz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307012

Hilal Yüce
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307070

Özet— Bu rapor, Yazılım Geliştirme Laboratuvarı – II dersi kapsamında gerçekleştirilen akademik personel başvuru portalı projesinin detaylarını içermektedir. Proje, akademik personel atama sürecini yönetmek amacıyla geliştirilmiştir. Adminlerin ilan oluşturabildiği, adayların başvuru yapabildiği, yöneticilerin süreçleri yönetebildiği ve jüri üyelerinin değerlendirme yapabildiği uçtan uca bir sistem sunmak hedeflenmiştir. MERN yığını kullanılarak full-stack geliştirme yapılmış, kullanıcı kimlik doğrulaması JWT ile sağlanmış ve veri yönetimi için RTK Query tercih edilmiştir. Ayrıca AWS S3 servisi entegrasyonu ile kullanıcı dosyalarının bulut üzerinde güvenli bir şekilde saklanması hedeflenmiştir. Ek olarak kayıt esnasında T.C. kimlik numarası doğrulama için Nüfus Vatandaşlık İşleri KPS servisinde yararlanılmıştır.

Anahtar Kavramlar—mern, react, node.js, express.js, mongodb, jwt, aws s3, rtk query

I. GİRİŞ

Akademik personel atama süreci, üniversiteler ve diğer eğitim kurumları için önemli bir yönetim alanıdır. Bu süreçlerin daha verimli ve düzenli bir şekilde yönetilebilmesi için dijital platformlar büyük kolaylık sağlamaktadır. Bu proje, akademik personel başvuru portalı oluşturmayı hedeflemektedir. Sistem, adminlerin ilan oluşturabileceği, adayların başvuru yapabileceği, yöneticilerin başvuruları yönetebileceği ve jüri üyelerinin değerlendirmeler yapabileceği kapsamlı bir yapıyı sunmaktadır.

Proje MERN (MongoDB, Express.js, React, Node.js) yığını kullanılarak tam yığın (full-stack) geliştirilmiş ve güvenli bir kullanıcı kimlik doğrulaması sağlamak için JSON Web Token (JWT) kullanılmıştır. Veri yönetimi ve API istekleri için Redux Toolkit Query (RTK Query) kullanılmış, uygulamanın performansı ve verimliliği artırılmıştır. Kullanıcı dosyalarının güvenli bir şekilde saklanabilmesi için AWS S3 bulut depolama servisi ile entegrasyon gerçekleştirilmiştir. Kayıt esnasında T.C. kimlik numarası doğrulaması yapmak için NVİ (Nüfus ve Vatandaşlık İşleri) servisinde faydalanılmıştır.

Çalışmada akademik personel atama sürecinin dijitalleşmesi ve daha verimli hale getirilmesi için bir çözüm sunmak amaçlanmaktadır.

II. SİSTEM TASARIMI

Uygulama üç temel katmandan oluşmaktadır: İstemci (frontend), sunucu (backend) ve veri tabanı + dosya depolama.

A. İstemci (Frontend)

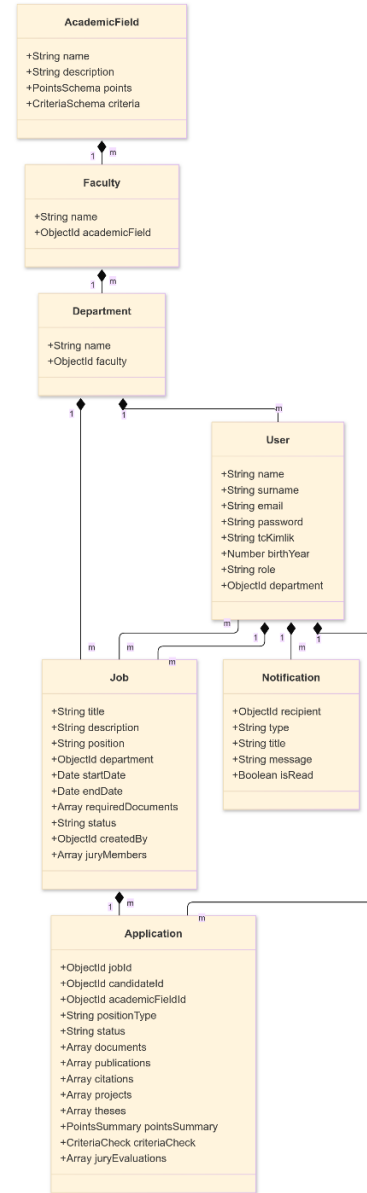
Kullanıcı arayüzü React[1] ile geliştirilmiştir. React içinde RTK Query[2] kullanılarak API istekleri yönetilmiş, veri çekme, gönderme ve cache işlemleri otomatik hale getirilmiştir. Kullanıcı login veya register olduğunda, backend tarafından verilen JWT Token[3] local storage üzerinde güvenli bir şekilde saklanmıştır.

B. Sunucu (Backend)

Node.js[4] ve Express[5] kullanılarak bir API sunucusu oluşturulmuştur. Kullanıcı kimlik doğrulama işlemleri burada gerçekleştirilmektedir. JWT tokenlar her korumalı API isteği öncesinde doğrulanarak, sadece yetkilendirilmiş kullanıcıların belirli kaynaklara erişmesine izin verilmiştir.

C. Veri Tabanı ve Dosya Depolama

Kullanıcı bilgileri ve uygulama verileri MongoDB[6] üzerinde saklanmıştır. Kullanıcılar tarafından yüklenen dosyalar AWS S3[7] servisine doğrudan gönderilmiştir. Bağlantılar veri tabanına kaydedilmiş ve erişim sağlanmıştır.



Görsel 1: MongoDB Koleksiyonları İlişki Diyagramı

Görsel 1'de projenin MongoDB veri tabanı yapısı gösterilmektedir. Diyagramda akademik personel başvuru sistemindeki temel veri koleksiyonları ve aralarındaki ilişkiler belirtilmiştir. Koleksiyonlar arasındaki bağlantılar referans ID'ler ile sağlanmıştır. Bu yapı, MongoDB'nin döküman odaklı NoSQL mimarisinin avantajlarını kullanarak hızlı sorgu performansı ve esnek şema yapısı sunmaktadır.

D. İşlem Akışı

1. Kullanıcı React arayüzü üzerinden istek gönderir.
2. RTK Query bu isteği API sunucusuna iletir.
3. API sunucusu JWT tokeni doğrular.
4. Geçerli istek olursa MongoDB'ye erişim veya AWS S3'e dosya yükleme yapılır.
5. Sonuçlar React tarafına geri döner ve kullanıcı arayüzü güncellenir.

HTTP Metodu	Bildirim API Rotaları		
	Endpoint	Controller Fonksiyonu	Açıklama
GET	/	getMyNotifications	Kullanıcının tüm bildirimlerini getirir
PUT	/:id/read	markAsRead	Belirli bir bildirimi okundu olarak işaretler
PUT	/read-all	markAllAsRead	Tüm bildirimleri okundu olarak işaretler
GET	/unread-count	getUnreadCount	Okunmamış bildirim sayısını döndürür

Tablo 1: Örnek API Rotaları - Bildirim Yönetimi

Tablo 1'de örnek olarak sistemdeki bildirim yönetimi için kullanılan API rotaları verilmiştir. Tüm rotalar kullanıcı kimlik doğrulaması gerektirmekte olup bildirim verilerinin güvenliği "protect" middleware'i ile sağlanmaktadır.

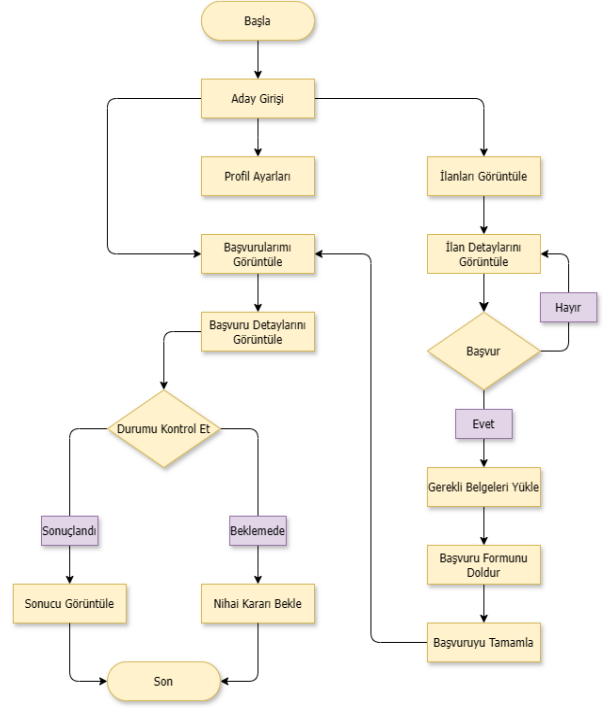
III. UYGULAMA

Bu proje, dört ana kullanıcı panelinden oluşmaktadır: aday, admin, yönetici ve jüri. Her bir panelin kendine özgü yetkileri ve işlevsellikleri vardır. Her panelin işlevselliği, kullanıcının sisteme erişim hakkı ve rolüne göre şekillenir. Kullanıcılar, sisteme kayıt olmadan önce NVİ KPS[8] servisi üzerinden ad, soyad, doğum yılı ve T.C. kimlik numarası kontrol edilerek kimlik doğrulamasından geçer ve doğrulama sonucunda sisteme kaydolurlar.

A. Aday Paneli

Aday paneli, başvuru yapmak isteyen kullanıcılar için tasarlanmıştır. Adaylar, bu panel üzerinden akademik personel ilanlarına başvurabilirler.

- **Başvuru Oluşturma:** Adaylar, aktif ilanları görüntüleyip başvuru yapabilirler. Başvuru yaparken istenen bilgileri doldurabilir ve belgeleri yükleyebilirler.
- **Başvuru Durumu:** Başvurdukları ilanların durumunu takip edebilirler. Başvuru sonucu hakkında bildirim alırlar.
- **Dosya Yükleme:** Başvurularına gerekli belgeleri ve dosyaları AWS S3 ortamına yükleyerek başvurularını tamamlarlar.



Görsel 2: Aday Paneli Akış Şeması

B. Admin Paneli

Admin paneli, sistemin genel yönetimi için gerekli tüm yetkilere sahip kullanıcılara sunulmaktadır. Adminler, başvuru sürecini yönetebilir, ilanlar oluşturabilir ve sistemdeki kullanıcıların rollerinde düzenlemeler yapabilir.

- **İlan Oluşturma:** Adminler, ilan oluşturabilir, düzenleyebilir veya silebilirler. İlanlara başvuru tarihleri, ilan açıklaması, istenen belgeler vb. bilgileri ekleyebilirler.

Görsel 2: Admin Paneli İlan Yönetimi Sayfası

- **Kullanıcı Yönetimi:** Aday ve diğer kullanıcıların sistemdeki hesaplarını yönetebilirler. Kullanıcıların rollerini belirleyebilirler.
- **Başvuru İnceleme:** Başvuruların detaylarını inceleyebilirler.

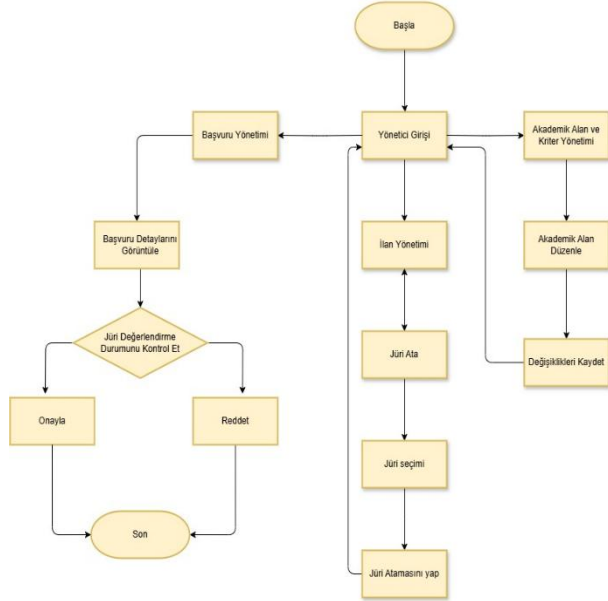
C. Yönetici Paneli

Yönetici paneli, süreci yöneten kişilere sunulmaktadır. Yöneticiler, atama sürecinin işleyişini izler, ilanlara jüri atayabilirler ve jüri değerlendirmeleri tamamlandığında başvurular hakkında nihai kararı verebilirler.

Başvuru Yönetimi							
Son Başvurular							
İlan, aday adı veya akademik alan ara...							
İlan Başlığı	Aday	Akademik Alan	Postayon	Tarih	Durum	Jüri Değerlendirmesi	İşlemler
Bilgi Sistemleri Mühendisliği Bölümü - Doktor Öğretim Üyesi Alımı	Aday 1 aday@edag.com	Sağlık Bilimleri, Fen Bilimleri ve Matematik, Mühendislik, Ziraat, Orman ve Su Ürünleri Temel Alanı	Yüksek Lisans	27.04.2025 23:01	Onaylandı	Değerlendirildi	İşlemler
Bilgi Sistemleri Mühendisliği Bölümü - Profesör Alımı	Aday 1 aday@edag.com	Sağlık Bilimleri, Fen Bilimleri ve Matematik, Mühendislik, Ziraat, Orman ve Su Ürünleri Temel Alanı	Yüksek Lisans	27.04.2025 23:00	Onaylandı	Değerlendirildi	İşlemler

Görsel 3: Yönetici Paneli Başvuru Yönetimi Sayfası

- **Bildirim:** Sisteme admin tarafından yeni bir ilan eklendiğinde ilgili ilana jüri ataması yapması için bildirim alır.
- **Başvuru İnceleme:** Başvuruların detaylarını görüntüleyebilir; puanlarına, jüri değerlendirmelerine ulaşır ve başvuruya özel PDF raporuna erişebilir.
- **Kriter Tanımlama:** Akademik alanlara göre başvuru kriterleri ve özel kadro şartlarını tanımlayabilir.
- **Nihai Karar:** Jüri değerlendirmeleri tamamlandığında başvuru için nihai kararı verir.



Görsel 4: Yönetici Paneli Akış Şeması

Yöneticiler, sistemin genel işleyişinin düzgün bir şekilde devam etmesini sağlamak için kritik bir rol üstlenirler. İlan sürecini takip eder, jüri değerlendirmelerini izler ve sonuçların doğruluğunu denetlerler. Aynı zamanda, akademik kadro ihtiyaçlarına göre başvuru kriterlerini belirleyerek sürecin şeffaf olmasını sağlar. Böylece tüm başvuru süreci etkin bir şekilde yönetilir ve gerektiğinde müdahale edilebilir.

D. Jüri Paneli

Jüri üyeleri, sistemdeki ilanların başvurularını değerlendirmek için özel yetkilere sahip kullanıcılar olarak

görev alırlar. Jüri üyelerinin rolü, başvuru sürecinin objektif ve şeffaf bir şekilde değerlendirilmesini sağlamaktır. Jüri panelindeki işleyiş şu şekilde gerçekleşir:

- **Bildirim:** Kendilerine atanan ilanlar hakkında bildirim alırlar. Bu bildirimler, jüri üyelerinin yalnızca kendilerine atanan başvuruları değerlendirmeleri gerektiğini hatırlatır.
- **İlan Başvurularını Görüntüleme ve Değerlendirme:** Jüri Üyeleri, kendilerine atanmış olan ilanlara ait başvuruları detaylı şekilde inceleyebilirler. Başvuruların her birini görüntüleyebilir, adayların sundukları belgeleri ve bilgileri değerlendirerek puanlama yapabilirler.

İlan Başvuruları							
Bilgi Sistemleri Mühendisliği Bölümü - Doktor Öğretim Üyesi Alımı							
Bilgi Sistemleri Mühendisliği - Son Başvuru: 27.05.2025 22:37							
Başvuru Listesi							
Aday	TC Kimlik	Toplam Puan	Kriter Kontrolü	Başvuru Durumu	Başvuru Tarihi	İşlemler	
Aday 1	44444444444	110	Kriter Kontrolü	Onaylandı	27.04.2025	İşlemler	

Görsel 5: Jüri Paneli İlan Başvuruları Sayfası

IV. PROJE YÖNETİMİ

Proje geliştirme sürecinde takım çalışması Notion platformu üzerinden yönetilmiştir. Süreç içerisinde; Notion'da oluşturulan çalışma alanında ekip üyelerinin görev dağılımları, kullanılacak kaynaklar, zaman planlaması ve ilerleme durumu takip edilmiştir.

Yazılım geliştirme sürecinde GitHub platformu kullanılarak versiyon kontrolü sağlanmıştır. Projenin kaynak kodlarına GitHub üzerinden erişilebilir:

- [GitHub Bağlantısı / Academic-Recruitment-Portal](#)

REFERANSLAR

- [1] React, "Installation - React" [Çevrimiçi]. Mevcut: <https://react.dev/learn/installation>. [Erişim Tarihi: 28-Nis-2025].
- [2] Redux Toolkit, "RTK Query Overview" [Çevrimiçi]. Mevcut: <https://redux-toolkit.js.org/rtk-query/overview>. [Erişim Tarihi: 28-Nis-2025].
- [3] JWT, "Introduction to JSON Web Tokens" [Çevrimiçi]. Mevcut: <https://jwt.io/introduction>. [Erişim Tarihi: 28-Nis-2025].
- [4] Node.js, "Node.js Documentation" [Çevrimiçi]. Mevcut: <https://nodejs.org/docs/latest/api/>. [Erişim Tarihi: 28-Nis-2025].
- [5] Express, "Express - Node.js web application framework" [Çevrimiçi]. Mevcut: <https://expressjs.com/>. [Erişim Tarihi: 28-Nis-2025].
- [6] MongoDB, "MongoDB Documentation" [Çevrimiçi]. Mevcut: <https://www.mongodb.com/docs/>. [Erişim Tarihi: 28-Nis-2025].
- [7] Amazon Web Services, "Amazon Simple Storage Service Documentation" [Çevrimiçi]. Mevcut: <https://docs.aws.amazon.com/s3/>. [Erişim Tarihi: 28-Nis-2025].
- [8] Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, "Kimlik Paylaşımı Sistemi" [Çevrimiçi]. Mevcut: <https://www.nvi.gov.tr/kps>. [Erişim Tarihi: 28-Nis-2025].