AKADEMİK PERSONAL BAŞVURU SİSTEMİ

Batuhan Demirbas   
Bilisim Sistemleri Muhendisligi  
Kocaeli,Izmit  
[demirbas\_batuhan@hotmail.com](mailto:demirbas_batuhan@hotmail.com)

Metin Batın Dincer   
Bilisim Sistemleri Muhendisligi  
Kocaeli,Izmit  
[dincermetin0@gmail.com](mailto:dincermetin0@gmail.com)

Süleyman Emir Kaya   
Bilisim Sistemleri Muhendisligi  
Kocaeli,Izmit  
[emirfb7@gmail.com](mailto:emirfb7@gmail.com)

ÖZET: Bu proje, üniversitelerde akademik personel alımı sürecinin dijitalleştirilmesi amacıyla tasarlanmış bir web uygulamasını içermektedir. Sistem, akademik ilanların yönetimi, başvuruların alınması, değerlendirilmesi ve sonuçlandırılması süreçlerini kapsamaktadır. Proje, React ve Node.js teknolojileri kullanılarak geliştirilmiş olup, veritabanı yönetimi için PostgreSQL kullanılmaktadır. Kullanıcı dostu arayüzü ve güvenli kimlik doğrulama sistemi ile üniversitelerin akademik işe alım süreçlerini modernize etmeyi hedeflemektedir.

# Giriş

Günümüzde üniversiteler, akademik personel alımı süreçlerinde geleneksel yöntemler kullanmaktadır. Bu süreçler genellikle kağıt bazlı başvurular, manuel değerlendirmeler ve yüz yüze görüşmeler içermektedir. Bu geleneksel yöntemler, zaman alıcı, maliyetli ve hata yapma olasılığı yüksek süreçlerdir. Bu projenin amacı, akademik personel alımı sürecini dijitalleştirerek daha verimli, şeffaf ve erişilebilir hale getirmektir.

#### Proje, aşağıdaki temel işlevleri içermektedir:

* - Akademik ilanların oluşturulması ve yönetimi
* - Başvuru kriterlerinin belirlenmesi
* - Adayların başvuru yapması ve belge yüklemesi
* - Başvuruların değerlendirilmesi
* - Jüri üyelerinin atanması ve değerlendirme yapması

Sistem, admin, yönetici, aday ve jüri üyesi olmak üzere dört farklı kullanıcı rolüne sahiptir. Her rol, sistemin belirli işlevlerine erişim sağlamaktadır.

# Sistem Mimarisi

## Genel Mimari

Proje, modern web uygulamalarında yaygın olarak kullanılan MERN (MongoDB, Express, React, Node.js) stack'inin bir varyasyonu olan PERN (PostgreSQL, Express, React, Node.js) stack'i üzerine inşa edilmiştir. Bu mimari, kullanıcı arayüzü (frontend) ve sunucu tarafı (backend) olmak üzere iki ana bileşenden oluşmaktadır.

## Frontend Mimarisi

Frontend, React kütüphanesi kullanılarak geliştirilmiştir. React, bileşen tabanlı bir yapı sağlayarak kodun tekrar kullanılabilirliğini artırmakta ve bakımını kolaylaştırmaktadır. Projede React Router kullanılarak sayfa yönlendirmeleri sağlanmıştır. Frontend'in yapısı şu şekildedir:

* **Components:** Tekrar kullanılabilir UI bileşenleri
* **Pages**: Sistemdeki ana sayfalar ve işlevler
* **Utils:** Yardımcı fonksiyonlar ve servisler
* **Assets:** Resimler, CSS dosyaları ve diğer statik dosyalar

## Backend Mimarisi

Backend, Node.js ve Express.js kullanılarak geliştirilmiştir. API tabanlı bir yapı ile frontend ile iletişim sağlamaktadır. Backend'in yapısı şu şekildedir:

* **Routes:** API endpoint'leri ve yönlendirmeleri
* **Middlewares:** Kimlik doğrulama ve hata yönetimi
* **DB Models**: Veritabanı tabloları ve ilişkileri
* **Controllers:** İş mantığı ve veri işleme

## Veritabanı Mimarisi

Veritabanı yönetim sistemi olarak PostgreSQL kullanılmıştır. ORM (Object-Relational Mapping) aracı olarak Sequelize kullanılarak veritabanı işlemleri yönetilmektedir. Ana veritabanı tabloları şunlardır:

* **Users:** Kullanıcı bilgileri
* **Ilanlar:** Akademik ilanlar
* **Basvurular:** Aday başvuruları
* **Belgeler:** Yüklenen belgeler
* **Kriterler:** İlan kriterleri
* **Puanlar:** Değerlendirme puanları
* **Ilan\_juri:** Jüri atama bilgileri

# Kullanılan Teknolojiler

## Frontend Teknolojileri

* **React:** Kullanıcı arayüzünün geliştirilmesi için kullanılan JavaScript kütüphanesi
* **React Router:** Sayfa yönlendirmeleri için kullanılan kütüphane
* **CSS:** Stil tanımlamaları için kullanılan dil
* **Vite:** Hızlı geliştirme ortamı sağlayan build aracı

## Backend Teknolojileri

* **Node.js:** Sunucu tarafı JavaScript çalıştırma ortamı
* **Express.js:** Web uygulama çerçevesi
* **Sequelize:** ORM aracı
* **JWT**: Kimlik doğrulama için JSON Web Token
* **Bcrypt:** Şifre şifreleme kütüphanesi
* **Multer**: Dosya yükleme işlemleri için kullanılan middleware

## Veritabanı Teknolojileri

* **PostgreSQL**: İlişkisel veritabanı yönetim sistemi

# Sistem İşlevleri

## Admin İşlevleri

* Yeni akademik ilan ekleme
* Ilanları güncelleme

## Yönetici İşlevleri

* İlan kriterlerinde değişiklik yapma
* Başvuru sürecini izleme
* Jüri üyelerini belirleme
* Nihai kararları verme

## Aday İşlevleri

* İlanları görüntüleme
* Başvuru yapma
* Belge yükleme
* Başvuru durumunu takip etme

## Jüri Üyesi İşlevleri

* Adayları değerlendirme
* Rapor yükleme

# Güvenlik Önlemleri

* **JWT Kimlik Doğrulama:** Kullanıcı oturumlarını güvenli bir şekilde yönetmek için JSON Web Token kullanılmaktadır.
* **Şifre Şifreleme:** Kullanıcı şifreleri, bcrypt algoritması kullanılarak şifrelenmektedir.
* **CORS Koruması:** Cross-Origin Resource Sharing (CORS) politikası ile yetkisiz erişimler engellenmektedir.
* **Dosya Doğrulama:** Yüklenen dosyaların türü, boyutu ve içeriği doğrulanmaktadır.

# Kullanıcı Arayüzü Tasarımı

## Ana Sayfa

Ana sayfa, sistemin genel bilgilerini ve kullanıcı girişi için bağlantılar içermektedir. Kullanıcılar, rollerine göre farklı giriş sayfalarına yönlendirilmektedir.

## İlan Yönetimi Arayüzü

Admin ve yöneticiler için tasarlanmış olan bu arayüz, ilanların oluşturulması, düzenlenmesi ve yönetilmesi işlevlerini sağlamaktadır.

## Başvuru Arayüzü

Adaylar için tasarlanmış olan bu arayüz, ilanları görüntüleme, başvuru yapma ve belge yükleme işlevlerini sağlamaktadır.

## Değerlendirme Arayüzü

Jüri üyeleri için tasarlanmış olan bu arayüz, adayları değerlendirme ve rapor yükleme işlevlerini sağlamaktadır.

# Sistem İşlevleri

1. Admin, yeni bir akademik ilan ekler.

2. Yönetici, ilan için başvuru kriterlerini belirler ve KOÜ Atama Yönetmeliği'ne uygunluk sağlar.

3. Aday, ilanı görüntüler ve başvuru yapar.

4. Sistem, adayın başvuruyu tamamlayıp tamamlamadığını kontrol eder.

5. Yönetici, ilan kriterlerinde değişiklik yapabilir.

6. Başvuru özeti, adayın başvurusunda sisteme otomatik olarak oluşturulur ve indirilebilir formatta sunulur.

7. Başvuru süresi tamamlandığında, yönetici başvuru sayısı hakkında bilgilendirilir.

8. Jüri üyeleri belirlenir ve her bir jüri üyesi değerlendirmesini yaparak raporunu sisteme yükler.

9. Tüm jüri üyelerinin değerlendirmeleri tamamlandıktan sonra, yönetici nihai kararı verir.

10. Sonuç, adaylara yönetici tarafından bildirilir.

# Test ve Değerlendirme

## Birim Testleri

Backend API'ların işlevselliğini test etmek için birim testleri yapılmıştır. Bu testler, API'ların beklenen davranışları sergileyip sergilemediğini kontrol eder.

## Entegrasyon Testleri

Frontend ve backend bileşenlerinin entegrasyonunu test etmek için entegrasyon testleri yapılmıştır. Bu testler, sistemin bütün olarak düzgün çalışıp çalışmadığını kontrol eder.

## Kullanıcı Kabul Testleri

Gerçek kullanıcılar ile sistemin kullanılabilirliğini ve işlevselliğini test etmek için kullanıcı kabul testleri yapılmıştır. Bu testler, sistemin kullanıcı beklentilerini karşılayıp karşılamadığını kontrol eder.

# Sonuç

Bu proje, akademik personel alımı sürecinin dijitalleştirilmesi amacıyla geliştirilmiş bir web uygulamasıdır. Proje, modern web teknolojileri kullanılarak geliştirilmiş olup, kullanıcı dostu arayüzü ve güvenli altyapısı ile üniversitelerin akademik işe alım süreçlerini modernize etmeyi hedeflemektedir.

##### References

1. React Dokümantasyonu. [Online]. Available: https://reactjs.org/docs/getting-started.html

2. Node.js Dokümantasyonu. [Online]. Available: https://nodejs.org/en/docs/

3. Sequelize Dokümantasyonu. [Online]. Available: https://sequelize.org/master/

4. Express.js Dokümantasyonu. [Online]. Available: https://expressjs.com/

5. PostgreSQL Dokümantasyonu. [Online]. Available: https://www.postgresql.org/docs/

6. JSON Web Token Dokümantasyonu. [Online]. Available: https://jwt.io/introduction/

7. React Router Dokümantasyonu. [Online]. Available: https://reactrouter.com/web/guides/quick-start

8. Vite Dokümantasyonu. [Online]. Available: https://vitejs.dev/guide/

9. "Web Application Development with MERN Stack", IEEE Conference on Web Services, 2022.

10. "Security Best Practices for Web Applications", Journal of Cybersecurity, Vol. 15, No. 3, 2023.