

Akademik Personel Başvuru Sistemi: Fakülte İstihdamı Yönetimi için Web Tabanlı Bir Çözüm

Esat Berat Uzunca
Yunus Emre Yılmaz
İbrahim Buğra San
Kocaeli Üniversitesi
221307054
221307044
221307059

Özet:

Bu makale, React, Next.js, Node.js, PostgreSQL, FastAPI ve Python gibi modern web teknolojileri kullanılarak geliştirilen bir Akademik Personel Başvuru Sistemi'nin tasarımını ve uygulamasını sunmaktadır. Sistem, akademik pozisyon ilanlarının, başvuru gönderimlerinin ve değerlendirme süreçlerinin yönetimini kolaylaştırmaktadır. Adaylar, yöneticiler, idareciler ve jüri üyeleri için rol tabanlı erişim sağlayarak, belge yükleme, otomatik puanlama ve değerlendirme raporlama gibi özellikler sunmaktadır. Sistemin modüler mimarisi, ölçeklenebilirlik ve değişen kurumsal gereksinimlere uyulanabilirlik sağlamaktadır.

I. GİRİŞ

Akademik kurumlar, fakülte işe alım süreçlerini yönetmek için verimli sistemlere ihtiyaç duymaktadır. Geleneksel kağıt tabanlı yöntemler zaman alıcı, hataya açık ve şeffaflıktan yoksundur. Bu makale, ilan duyurusundan nihai seçime kadar tüm işe alım iş akışını kolaylaştırmak için tasarlanmış web tabanlı bir Akademik Personel Başvuru Sistemi'ni açıklamaktadır.

Sistem, akademik işe alımdaki çeşitli zorlukları ele almaktadır:

- Farklı gereksinimlere sahip birden fazla pozisyon ilanını yönetmek
- Başvuru belgelerini toplamak ve düzenlemek
- Adayları kuruma özgü kriterlere göre

değerlendirmek

- Jüri üyesi incelemelerini koordine etmek
- Kurumsal düzenlemelere uygunluğu sağlamak

II. SİSTEM MİMARİSİ

Akademik Personel Başvuru Sistemi, açıkça ayrılmış ön uç ve arka uç bileşenleri ile modern bir web mimarisi kullanmaktadır.

A. Ön Uç Mimarisi

Ön uç, duyarlı ve etkileşimli bir kullanıcı arayüzü sağlayan React ve Next.js kullanılarak oluşturulmuştur. Sistem, Next.js uygulaması içinde çeşitli React bileşenlerini içermektedir. Bu bileşenler, sayfalar/rotalar, UI bileşenleri ve durum yönetimi olarak kategorize edilebilir. Kullanıcı arayüzleri, aday arayüzü, admin arayüzü, yönetici arayüzü ve jüri arayüzü olmak üzere dört ana kategoriye ayrılmıştır.

React, yeniden kullanılabilir UI öğelerine ve verimli işlemeye olanak tanıyan bileşen tabanlı mimarisi nedeniyle seçilmiştir. Next.js, ilk yükleme performansını ve arama motoru optimizasyonunu iyileştiren sunucu tarafı işleme yetenekleri sağlamaktadır.

B. Arka Uç Mimarisi

Arka uç iki ana bileşenden oluşmaktadır:

1. ****Node.js Sunucusu****: Kimlik doğrulama, kullanıcı yönetimi ve genel uygulama mantığını işler. Bu bileşen, kimlik doğrulama, kullanıcı yönetimi ve ilan yönetimi gibi temel işlevleri sağlar.
2. ****FastAPI (Python)****: Karmaşık hesaplamaları, belge işlemeyi ve özel algoritmaları yönetir. Bu bileşen, belge işleme, puan hesaplama ve rapor oluşturma gibi özelleştirilmiş işlevleri yerine getirir.

Her iki bileşen de PostgreSQL veritabanına bağlanır ve istemci isteklerini API geçidi aracılığıyla alır.

C. Veritabanı Tasarımı

PostgreSQL, sağlamlığı, karmaşık sorguları desteklemesi ve ilişkisel verileri verimli bir şekilde işleme yeteneği nedeniyle veritabanı yönetim sistemi olarak seçilmiştir. Veritabanı şeması, kullanıcılar, roller, ilanlar, pozisyonlar, başvurular, belgeler, kriterler, değerlendirmeler ve jüri üyeleri için tablolar içermektedir. Tablolar arasındaki ilişkiler, yabancı anahtar kısıtlamaları ile korunmaktadır.

III. SİSTEM İŞLEVSELLİĞİ

A. Kullanıcı Roller ve Erişim Kontrolü

Sistem dört farklı kullanıcı rolü uygulamaktadır:

1. ****Aday****: Akademik pozisyonlara başvuran bireyler
2. ****Yönetici****: İlanları oluşturmak ve yönetmekten sorumlu personel
3. ****İdareci****: Başvuru kriterlerini tanımlayan ve değerlendirme sürecini denetleyen yetkililer
4. ****Jüri Üyesi****: Başvuruları değerlendiren ve değerlendirme raporu sağlayan fakülte üyeleri

Her rol, JWT kimlik doğrulama ve rol tabanlı erişim kontrolü kullanılarak uygulanan sistem içinde belirli izinlere ve erişim seviyelerine sahiptir.

B. İlan Yönetimi

Yöneticiler akademik pozisyonlar için ilan oluşturabilir, düzenleyebilir ve yayınlatabilir. Her ilan şunları içerir:

- Pozisyon detayları (unvan, bölüm, fakülte)
- Başvuru zaman çizelgesi (başlangıç ve bitiş tarihleri)
- Gerekli nitelikler ve belgeler
- Pozisyona özgü değerlendirme kriterleri

C. Başvuru Süreci

Adaylar sisteme TC kimlik numarası ve şifre kullanarak erişir. Başvuru süreci şunları içerir:

1. Mevcut ilanları görüntüleme
2. Başvurulacak bir pozisyon seçme
3. Gerekli belgeleri yükleme (yayınlar, sertifikalar vb.)
4. Akademik başarılar hakkında bilgi sağlama
5. Başvuruyu gönderme

Sistem, gönderilenleri pozisyona özgü gerekliliklere göre doğrular ve adaylara eksik bilgiler hakkında bildirim yapar.

D. Değerlendirme İş Akışı

Değerlendirme iş akışı şu adımları içerir:

1. Başvuru gönderimi
2. İlk doğrulama
3. İdareci tarafından jüri atanması
4. Jüri değerlendirmesi
5. Bireysel raporların hazırlanması

6. İdareci incelemesi

7. Nihai kararın verilmesi

8. Adaya sonucun bildirilmesi

IV. UYGULAMA DETAYLARI

A. Ön Uç Uygulaması

Ön uç, navigasyon için App Router kullanan React 18 ve Next.js 14 kullanılarak uygulanmıştır. Temel uygulama yönleri şunlardır:

1. ****Bileşen Yapısı****: Yeniden kullanılabilirlik ve bakım kolaylığı için modüler bileşenler
2. ****Durum Yönetimi****: Global durum için Context API ve sunucu durumu için React Query
3. ****Form İşleme****: Doğrulama için Zod ile React Hook Form
4. ****UI Tasarımı****: Tailwind CSS kullanarak duyarlı tasarım

B. Arka Uç Uygulaması

Arka uç iki ana hizmetten oluşmaktadır:

1. ****Node.js API (Express)****:
 1. Kullanıcı kimlik doğrulama ve yetkilendirme
 2. İlanlar ve başvurular için CRUD işlemleri
 3. Dosya yükleme yönetimi
 4. Bildirim sistemi
2. ****FastAPI Hizmeti****:
 1. Belge işleme ve metin çıkarma
 2. Yayın metriklerine dayalı otomatik puanlama
 3. PDF formatında rapor oluşturma
 4. Harici API'lerle entegrasyon (e-Devlet hizmetleri)

C. Veritabanı Uygulaması

PostgreSQL veritabanı şeması şunlar için tablolar içermektedir:

- Kullanıcılar ve roller
- İlanlar ve pozisyonlar
- Başvuru belgeleri
- Değerlendirme kriterleri
- Jüri atamaları ve raporları

Tablolar arasındaki ilişkiler yabancı anahtarlar aracılığıyla korunmakta ve performansı optimize etmek için sık sorgulanan alanlar için indeksler oluşturulmaktadır.

D. Entegrasyon Noktaları

Sistem çeşitli harici hizmetlerle entegre olmaktadır:

1. ****Belge Depolama****: Güvenli belge depolama için AWS S3
2. ****E-posta Bildirimleri****: E-posta bildirimleri için SMTP hizmeti
3. ****Kimlik Doğrulama****: Kimlik doğrulama için e-Devlet API'si
4. ****PDF Oluşturma****: Rapor oluşturma için PDF oluşturma kütüphanesi

V. GÜVENLİK HUSUSLARI

Sistemde uygulanan güvenlik önlemleri şunları içermektedir:

1. ****Kimlik Doğrulama****: Yenileme belirteçleri ile JWT tabanlı kimlik doğrulama
2. ****Yetkilendirme****: Tüm sistem işlevleri için rol tabanlı erişim kontrolü
3. ****Veri Koruması****: Hassas verilerin aktarım sırasında ve beklerken şifrelenmesi
4. ****Girdi Doğrulama****: Tüm kullanıcı girdilerinin kapsamlı doğrulaması
5. ****Denetim Günlüğü****: Tüm sistem faaliyetlerinin ayrıntılı günlüğü

VI. TEST VE DOĞRULAMA

Sistem kapsamlı testlerden geçmiştir:

1. ****Birim Testi****: İzole olarak test edilen bireysel bileşenler
2. ****Entegrasyon Testi****: Bileşen etkileşimlerinin testi
3. ****Uçtan Uca Test****: Tam kullanıcı iş akışlarının testi
4. ****Performans Testi****: Yük altında sistem davranışı
5. ****Güvenlik Testi****: Güvenlik açığı değerlendirme ve penetrasyon testi

VII. SONUÇLAR VE TARTIŞMA

Akademik Personel Başvuru Sistemi, akademik işe alım süreçlerini yönetme zorluklarını başarıyla ele almaktadır. Temel faydalar şunlardır:

1. ****Verimlilik****: İdari yük ve işlem süresinde azalma
2. ****Şeffaflık****: Başvuru ve değerlendirme sürecinde net görünürlük

3. ****Tutarlılık****: Standartlaştırılmış değerlendirme kriterleri ve puanlama

4. ****Erişilebilirlik****: Tüm paydaşlar için uzaktan erişim

5. ****Veri Bütünlüğü****: Doğru kayıt tutma ve raporlama

VIII. SONUÇ VE GELECEK ÇALIŞMALAR

Bu makale, React, Next.js, Node.js, PostgreSQL, FastAPI ve Python kullanarak bir Akademik Personel Başvuru Sistemi'nin tasarımını ve uygulamasını sunmuştur. Sistem, ilan duyurusundan nihai seçime kadar akademik işe alım süreçlerini yönetmek için kapsamlı bir çözüm sunmaktadır.

Gelecek çalışmalar şunlara odaklanacaktır:

1. İşe alım eğilimleri için analitik yeteneklerin geliştirilmesi
2. Ön başvuru taraması için makine öğrenimi uygulaması
3. Otomatik doğrulama için akademik veritabanlarıyla entegrasyonun genişletilmesi
4. Gelişmiş erişilebilirlik için mobil uygulamaların geliştirilmesi

KAYNAKLAR

- [1] React Dokümantasyonu, "React ile Başlarken," [Çevrimiçi]. Erişilebilir: <https://reactjs.org/docs/getting-started.html>
- [2] Next.js Dokümantasyonu, "Başlarken," [Çevrimiçi]. Erişilebilir: <https://nextjs.org/docs>
- [3] Node.js Dokümantasyonu, "Node.js Hakkında," [Çevrimiçi]. Erişilebilir: <https://nodejs.org/en/about/>
- [4] PostgreSQL Dokümantasyonu, "PostgreSQL Nedir?," [Çevrimiçi]. Erişilebilir: <https://www.postgresql.org/about/>
- [5] FastAPI Dokümantasyonu, "FastAPI Framework," [Çevrimiçi]. Erişilebilir: <https://fastapi.tiangolo.com/>
- [6] Python Software Foundation, "Python Dil Referansı," [Çevrimiçi]. Erişilebilir: <https://docs.python.org/3/reference/>

[7] IEEE, "IEEE Konferans Şablonları," [Çevrimiçi].

Erişilebilir:

<https://www.ieee.org/conferences/publishing/templates.html>