

Akademik Kadro Alım Otomasyonu

Ad Soyad: Ahmet Salih Kocatürk
Numara: 221307043
Görev Alanları: Kullanıcı Giriş -
Kimlik Doğrulama - Aday ve Admin
Panellerinin Tasarımı ve Entegrasyonu
- Server Kodu

Ad Soyad: Anıl Ceco
Numara: 221307047
Görev Alanları: Yönetici ve Jüri
Panellerinin Tasarımı ve Entegrasyonu
- Server Kodu

Ad Soyad: Emircan Demir
Numara: 241307109
Görev Alanları: SQL Veritabanının
Projeye Yönelik Tasarlanması

I. GİRİŞ

Bu projede üniversite alanları için farklı kadro türleri üzerinden akademik kadro alım sistemi hedeflenmiştir.

Yükseköğretim kurumlarında akademik personel alım süreçleri, hem adaylar hem de değerlendirme kurulları açısından oldukça hassas, zaman alıcı ve bürokratik işlemlerle doludur. Bu süreçlerin dijital ortama taşınarak daha şeffaf, hızlı ve izlenebilir hale getirilmesi gerekliliği, modern üniversite yönetim sistemlerinin önemli bir ihtiyacı haline gelmiştir. Bu doğrultuda geliştirilen Akademik Personel Başvuru Sistemi, Kocaeli Üniversitesi Akademik Atama Yönergesi esas alınarak, öğretim üyeliği kadrolarına yapılacak başvuruların elektronik ortamda yönetilmesini sağlayan kapsamlı bir dijital çözümdür. Sistem; adayların başvuru yapabildiği, yöneticilerin başvuru kriterlerini tanımlayabildiği, jüri üyelerinin değerlendirme yapabildiği ve admin kullanıcıların ilan süreçlerini yönetebildiği dört ana kullanıcı rolü etrafında inşa edilmiştir. Her rolün kendine özgü işlevleri, ekranları ve yetkileri sistem içinde detaylı biçimde tanımlanmıştır. Adaylar, kişisel bilgileri ve belgeleri ile ilanlara başvuru yapabilir, bilimsel faaliyetlerini sistem üzerinden beyan ederek otomatik puanlama alabilir ve tüm başvuru sürecini anlık olarak takip edebilir. Yöneticiler, ilana özel kriterleri girerek başvuruların yönergeye uygun biçimde değerlendirilmesini sağlar. Jüri üyeleri ise yalnızca atandıkları başvuruları görebilir ve değerlendirme raporlarını sisteme yükleyebilir. Admin kullanıcılar ise ilan ekleme, düzenleme ve sistem yönetimi gibi temel işlemleri gerçekleştirir. Böylece sistem, akademik personel alım sürecini hem kullanıcı dostu hem de mevzuata uygun bir şekilde dijitalleştirerek; insan hatalarını en aza indirir, belge takibini kolaylaştırır ve değerlendirme süreçlerini daha objektif hale getirir. Geliştirilen bu sistem, yalnızca bir başvuru aracı değil; aynı zamanda üniversitelerin dijital dönüşüm vizyonunu destekleyen yenilikçi bir altyapıdır.

II. KULLANILAN TEKNOLOJİLER

Akademik Portal Sistemi, modern ve kullanıcı dostu bir yapı sunmak amacıyla Flutter, Node.js ve MySQL teknolojileri temelinde geliştirilmiştir. Ön yüz (frontend) tarafında, mobil ve web uyumlu arayüzler oluşturmak için Google'ın açık kaynaklı Flutter çerçevesi tercih edilmiştir. Uygulama durumu yönetimi için ise BLoC (Business Logic Component) mimarisi kullanılmış, her kullanıcı rolüne özgü Cubit ve State yapıları oluşturulmuştur. Arka uç (backend) tarafında Express.js çatısı altında Node.js kullanılarak RESTful API servisleri geliştirilmiştir. Veriler ise ilişkisel bir yapı ile MySQL veritabanında saklanmaktadır. Bununla birlikte, kullanıcı girişleri, ilan yönetimi, başvuru süreci ve jüri değerlendirmeleri gibi süreçler tamamen özelleştirilmiş API'ler aracılığıyla gerçekleştirilmiştir. Ayrıca başvuru sürecinin sonunda, başvuru faaliyetlerini ve puanlarını içeren resmi bir "Tablo 5" belgesi pdf ve printing kütüphaneleriyle Flutter içinde PDF formatında otomatik olarak oluşturulmaktadır.

III. BACK-END ALTYAPISI

Bu projede backend altyapısı, modern web geliştirme standartlarına uygun olarak Node.js ve Express.js çerçevesinde yapılandırılmıştır. Geliştirilen RESTful API servisleri, Flutter tabanlı istemcilerden gelen istekleri işleyerek sistemin tüm iş mantığını yerine getirir. Veritabanı işlemleri, mysql2 kütüphanesi ile gerçekleştirilmekte ve tüm CRUD işlemleri bu yapı üzerinden yürütülmektedir. Sunucu tarafında çevresel değişkenler .env dosyası ile tanımlanmış, böylece bağlantı bilgileri ve port gibi hassas veriler koddan izole edilmiştir. Proje içerisinde yer alan index.js dosyası, sunucunun başlatılmasını, veritabanı bağlantısını ve tüm API endpoint'lerinin tanımlanmasını sağlar.

Backend, kullanıcılar, ilanlar, başvurular, kadrokriterleri ve juriatamaları gibi veritabanı tabloları üzerinde işlem yapmaktadır. Kullanıcı girişleri /login endpoint'i aracılığıyla gerçekleştirilir ve gelen TC Kimlik No ile şifre bilgisi doğrudan veritabanındaki kullanıcılar tablosuyla eşleştirilir. Eğer bilgiler eşleşirse, kullanıcıya ait ad, soyad, rol ve TC bilgisi JSON formatında geri döndülür. Giriş yapılan her kullanıcı, sistem içinde sahip olduğu RolID değerine göre frontend tarafında yönlendirilerek ilgili ekrana ulaşır.

Kayıt işlemi ise /register endpoint'i ile gerçekleştirilmekte olup, SOAP servisi ile kimlik doğrulaması yapılmış olsa dahi olmasa dahi veritabanına kayıt atılmaktadır. auth_cubit.dart içerisinde yer alan registerWithSOAP() fonksiyonu, önce SOAP servisinden dönüş beklemekte, daha sonra bu bilgileri içeren POST isteğini veritabanına yönlendirmektedir. Böylece dış servislerin erişilemediği durumlarda dahi kullanıcı kaydı işlemi aksamadan sürdürülmektedir.

İlan yönetimi işlemleri /announcements endpoint'leri üzerinden yapılır. Yeni ilan eklemek için POST, tüm ilanları getirmek için GET, silmek için DELETE, güncellemek için ise PUT metotları kullanılmıştır. Başvurular da benzer şekilde /applications endpoint'i üzerinden erişilir ve tüm başvurular listelenebilir ya da yeni başvurular alınabilir. Yöneticilerin tanımladığı akademik kriterler kadrokriterleri tablosuna kayıt edilir ve bu kayıtlar hem yönetici panelinde hem de jüri değerlendirmesinde kullanılmak üzere API ile servis edilir.

Ayrıca jüri üyelerinin belirli ilanlara atanmasını sağlayan /juri-atama endpoint'i, sistemin değerlendirme süreci için hayati önem taşır. Burada, TC Kimlik Numarasına göre kullanıcının KullanıcıID'si bulunur ve ilanla ilişkilendirilerek juriatamaları tablosuna kayıt yapılır. Jüri üyeleri daha sonra yalnızca atandıkları başvurulara erişebilir ve değerlendirme işlemlerini /juri-basvurular ve /juri-degerlendirme gibi özel uç noktalar üzerinden gerçekleştirir.

Tüm bu yapı, kullanıcıdan gelen verilerin doğruluğunu sağlayan, rol tabanlı erişim sunan ve farklı iş akışlarına göre özelleştirilen güçlü bir REST API mimarisi ile tamamlanmıştır. Sistemin ölçeklenebilirliği göz önünde bulundurularak modüler endpoint yapısı tercih edilmiş ve her işlem için özel olarak hazırlanmış sorgular kullanılmıştır. Bu

sayede, uygulama veri bütünlüğünü ve işlem güvenliğini yüksek oranda koruyabilmektedir.

IV. GİRİŞ VE AUTHENTICATION

A. Rol Bazlı Giriş Ekranı

Sistemde dört ana kullanıcı rolü bulunduğundan, her bir rol için özel olarak tasarlanmış giriş ekranları hazırlanmıştır. Aday, jüri, yönetici ve admin kullanıcıları kendi giriş sayfalarından sisteme erişim sağlar. Bu ekranlarda kullanıcıdan TC Kimlik Numarası ve şifre bilgisi alınır. Giriş işlemi AuthCubit sınıfındaki login() fonksiyonu ile gerçekleştirilir. Girilen bilgiler HTTP POST yöntemiyle backend'e gönderilir ve doğrulama başarılı olursa, kullanıcının rolü belirlenerek ilgili panel ekranına yönlendirme yapılır. Bu yönlendirmeler sayesinde, her kullanıcı sadece kendi yetkisine uygun bölümleri görebilir ve kullanabilir.

B. Kayıt Olma Süreci

Giriş ekranlarında ayrıca yeni kullanıcı kaydı için "Kayıt Ol" butonuna yer verilmiştir. Bu butona tıklandığında açılan form üzerinden kullanıcıdan çeşitli bilgiler talep edilir. TC Kimlik No, ad, soyad, doğum yılı, e-posta adresi ve şifre bilgileri eksiksiz olarak girildikten sonra kayıt işlemi başlatılır. Formun gönderilmesiyle birlikte bu veriler registerWithSOAP() fonksiyonuna iletilir ve sistem hem kimlik doğrulaması hem de veritabanı kaydı için gerekli işlemleri başlatır. Her rol için tanımlı rolId değeri de bu aşamada backend'e gönderilerek kullanıcının sistemi hangi yetkiyle kullanacağı belirlenmiş olur.

C. Soap Servisiyle Kimlik Doğrulama denemesi

Kayıt işlemi sırasında, kullanıcıdan alınan TC Kimlik Numarası ve diğer kimlik bilgileri bir SOAP servisine yönlendirilerek gerçek kişi doğrulaması yapılmaya çalışılır.

D. Doğrulama Başarısız Olsa bile Kayıt Tamamlanır

Ancak geliştirme sürecinde SOAP servisine doğrudan erişim mümkün olmadığından dolayı, servis yanıt veremese bile sistemin çalışmaya devam etmesi sağlanmıştır. Yani SOAP doğrulaması başarısız olsa dahi kullanıcı bilgileri MySQL veritabanına kaydedilir. Bu sayede internet bağlantısı kısıtlı ortamlarda ya da test ortamında çalışan geliştiriciler için sistemin işlevselliği korunmuş olur. Kullanıcıya kayıt işlemi sonunda, kimliğin doğrulanıp doğrulanmadığı açık bir mesajla belirtilir. Bu mesajlar, "Kimlik doğrulandı ve kayıt başarılı!" ya da "SOAP doğrulaması yapılamadı ama kayıt başarılı." şeklinde kullanıcıyı bilgilendirecek şekilde tasarlanmıştır. ve kullanıcıya pozitif geri bildirim sağlar. Bu işlem uygulamanın gerçek dünya entegrasyon kabiliyetini ortaya koyan önemli bir adımdır.

E. Teknik Akışın Esnekliği

Bu yapı sayesinde sistem hem kamu servislerine entegre olacak biçimde ileriye dönük hazır hale getirilmiş, hem de esnekliği sayesinde her ortamda sorunsuz çalışabilecek şekilde yapılandırılmıştır. Giriş ve kayıt işlemleri, kullanıcı açısından kesintisiz ve akıcı bir deneyim sunarken, geliştirici tarafında da maksimum uyumluluk sağlanmıştır. SOAP servisine ulaşmanın mümkün olmadığı durumlarda bile veritabanı kaydının gerçekleşmesi, uygulamanın sağlam bir

yapıda kurgulandığını ve gerçekçi senaryolara uygun biçimde tasarlandığını göstermektedir.

V. ROLLER

A. Aday

Aday rolü, sistemin başvuru tarafındaki temel kullanıcı kitlesini temsil eder. Adaylar, role selector ekranından "Aday Girişi" seçeneğini kullanarak candidate_login.dart ekranı üzerinden sisteme giriş yaparlar. Giriş sırasında TC Kimlik Numarası ve şifre bilgisi alınır; kullanıcı bu bilgileri girdikten sonra AuthCubit aracılığıyla doğrulama yapılır ve başarılı giriş durumunda /home/candidate rotasına yönlendirilir. Aday giriş yaptıktan sonra CandidateHome sayfası açılır. Bu sayfada, adaylara iki sekme sunulur: Duyurular ve Başvurularım. Duyurular sekmesinde, sistemde yayınlanmış tüm aktif ilanlar listelenir ve her bir ilanın detayları (kadro tipi, temel alan, tarih aralığı, gerekli belgeler ve açıklama) gösterilir. Adaylar, bu ekran üzerinden başvuru işlemi gerçekleştirebilirler.

Başvuru işlemi sırasında aday, sistem tarafından belirlenen ilan kriterlerini görür ve faaliyetlerini belirli puanlara göre girer. candidate_application.dart dosyasında tanımlanan başvuru ekranı, adayın makale, kitap, bildiri, ödül, idari görev gibi çeşitli bilimsel faaliyetlerini tiplerine göre seçmesine ve bu faaliyetlerin detaylarını girerek puan almasına olanak sağlar. Faaliyetler sistem tarafından otomatik puanlandırılır ve adayın toplam puanı hesaplanır. Bu işlemlerin ardından aday, başvurusunu tamamlayarak PDF çıktısı alabilir. Adayın daha önce yaptığı başvurular ise "Başvurularım" sekmesinde listelenir. Burada her başvuruya ait tarih, başvuru durumu ve ilgili ilan bilgisi gösterilir. Bu yapı, adayların başvuru sürecini hem etkili hem de izlenebilir bir biçimde yönetmelerine olanak tanır.

B. Admin

Admin rolü, sistemdeki ilanların yönetiminden sorumlu olan kullanıcıları ifade eder. Admin giriş ekranı admin_login.dart içerisinde tanımlanmıştır ve tıpkı diğer roller gibi TC Kimlik Numarası ve şifre ile giriş yapılır. Başarılı bir giriş sonrasında kullanıcı, AdminHome ekranına yönlendirilir. Bu panelde admin kullanıcılara iki ana sekme sunulur: İlanları Görüntüle ve Yeni İlan Ekle. İlanları Görüntüle sekmesinde, sistemdeki tüm ilanlar listelenir. Her ilanın başlığı, tarih aralığı ve işlem butonları (düzenle, sil) yer alır. Admin kullanıcıları, mevcut ilanları düzenleyebilir veya silebilir. Düzenleme ekranında, ilan başlığı, açıklaması, başlangıç ve bitiş tarihi, gerekli belgeler, kadro tipi ve temel alan bilgileri güncellenebilir.

Yeni ilan eklemek isteyen admin, "Yeni İlan Ekle" sekmesine geçerek ilgili formu doldurur. Bu formda, ilan ile ilgili tüm bilgiler kullanıcıdan alınır ve kaydedilir. Kadro tipi ve temel alan seçimleri sayısal ID üzerinden yapılmakta, fakat sistem tarafından metin olarak gösterilmektedir. Admin bir ilan oluşturduğunda, bu ilan tüm aday kullanıcılar tarafından görüntülenebilir ve başvuruya açık hale gelir. Admin paneli sade ama işlevsel bir yapıya sahiptir ve sistemdeki tüm ilanların merkezden kontrol edilmesini sağlar.

C. YÖNETİCİ

Yönetici rolüne sahip kullanıcıyı, giriş işlemlerinden sonra 3 farklı panel karşılamaktadır. Bu paneller sırasıyla "Kriter Düzenleme", "Jüri Atama", "Başvuru Değerlendirme".

Kriter Düzenleme: Bu ekran, yöneticinin her akademik kadro (Dr. Öğr. Üyesi, Doçent, Profesör) için başvuru değerlendirme kriterlerini tanımlamasını sağlar.

Genel kriter belirlenmesine yönelik tasarlanan bu alanda Tablo-1'de bulunan verilerden yola çıkılarak işlemler yapılmaktadır. Örneğin Mühendislik alanında profesör kadrosu için kriter belirlenirken ilgili makale aralıkları (A1-A4, A1-A6...) seçimi yapılır, seçilen makale için gerekli adet ve bu alanı sağlayacağı takdirde getirmesi beklenen puan miktarı yazılır.

Alınması gereken minimum puan değeri de girildiğinde çoklu bir şekilde kriter eklemesi yapılır. Eklenen kriterler doğrudan veritabanına kaydedilir.

Eklenen kriterlerin puanlamaları sonradan güncellenebilir yapıdadır. Aynı zamanda seçili bir kriter de silinebilmektedir.

Jüri Atama: Veritabanı üzerinden kayıtlı ilanlar yönetici paneline direkt çekilip UI tarafında gösterilmektedir. Her bir ilan için 5 farklı atama alanı bulunmaktadır. Her bir alan, veritabanına gönderilen istek doğrultusunda UI tarafında doldurulmaktadır. “Jüri Üyesi” rolünden kayıtlı olan kullanıcılar, atama alanlarında listelenir. 5 farklı üye seçiminin ardından atama işlemi gerçekleşir.

Başvuru Değerlendirme: Kayıtlı ilanlara gelen başvurular, yönetici olan kullanıcının ekranına veritabanı üzerinden çekilmektedir. Yönetici, bu alanda gelen başvurular için nihai kararı belirler ve olumlu ya da olumsuz olarak seçim yapar. Yapılan seçim, aynı zamanda veritabanında güncellenir.

C. JÜRİ ÜYESİ

Jüri üyesi olarak sisteme giriş yapan kullanıcı, veritabanı üzerinden çekilen bilgiler doğrultusunda kendi TC ve id değerine atandığı ilanların başvurularını panelde görmektedir.

Her bir başvuru için bir değerlendirme kutusu açılır. Jüri, burada başvuran aday için bir değerlendirme notu yazar ve kendi yetkisi dahilinde karar işlemini gerçekleştirir. İşlem tamamlandıktan sonra kendi ID değerine yönelik veritabanında “değerlendirme” başlığı altında jürinin değerlendirmesi kaydedilir.

VI. SİSTEM ARAYÜZÜ

Yönetici Paneli:

Jüri Atama: Yöneticinin, açılan ilanlar için jüri rolünde sistemde kayıtlı olan kullanıcılar üzerinden atama yaptığı panelidir.

Bilgisayar Mühendisliği Bölümü Dr. Öğr. Üyesi Kadrosu
Açıklama: Bilgi Sistemleri Anabilim Dalı için Dr. Öğr. Üyesi alımı yapılacaktır.
Başlangıç Tarihi: 2025-04-30T21:00:00.000Z
Bitti Tarihi: 2025-05-30T21:00:00.000Z
Jüri 1
Üye Seçin
Jüri 2
Prof. Dr. Zeynep Şahin (55555555555)
Doç. Dr. Hasan Öztürk (66666666666)
Ali Veli (77777777777)
Anıl Ceko (88888888888)
Yiğit Yıldırım (12345678901)
Ahmet Salih Kocatürk (98765432101)
Emircan Demir (12345678901)

Kriter Ekleme: Yöneticinin belirli bir alan için Öğretim Üyesi, Doçent ve Profesör kadrolarına yönelik kriter belirlediği panelidir.

Mühendislik
Doçent
Min Puan: 30 A1-A4: Adet 3, Max Puan: 60
Min Puan: 30 A1-A5: Adet 3, Max Puan: 90
Fen Bilimleri ve Matematik
Profesör
Min Puan: 30 A1-A4: Adet 3, Max Puan: 90
Min Puan: 30 E1-E4: Adet 4, Max Puan: 100
Min Puan: 30 H1-H17: Adet 2, Max Puan: 100
Kriter Belirleme
Jüri Atama

Başvuru Nihai Değerlendirme: Yöneticinin başvurular için son kararı verdiği alandır.

İlan İnceleme
Ayşe Yılmaz - Bilgisayar Mühendisliği Bölümü Dr. Öğr. Üyesi Kadrosu
Başvuru Tarihi: 2025-04-25T14:52:13.000Z
İlan Süresi: 2025-04-30T21:00:00.000Z - 2025-05-30T21:00:00.000Z
Durum: Reddedildi
Toplam Puan: 0
PDF İndir
Mehmet Kaya - Bilgisayar Mühendisliği Bölümü Dr. Öğr. Üyesi Kadrosu
Başvuru Tarihi: 2025-04-25T14:52:13.000Z
İlan Süresi: 2025-04-30T21:00:00.000Z - 2025-05-30T21:00:00.000Z
Durum: Onaylandı
Toplam Puan: 0
PDF İndir
Nihai Kararı Belirle
<input checked="" type="radio"/> Onaylandı
<input type="radio"/> Reddedildi
İptal Kaydet

Admin Paneli:

İlan Ekleme: Admin rolündeki kullanıcının farklı kategoriler için ilan açabildiği panelidir.

Admin Paneli
Sistem Yönetimi
Yeni Başvuru
Açıklama
Başlangıç Tarihi
Bitti Tarihi
Genel Bilgi (Görünür)
Kullanıcı Seçin
Tamamla
İlan Ekleme
Yeni Başvuru
Açıklama
Başlangıç Tarihi
Bitti Tarihi
Genel Bilgi (Görünür)
Kullanıcı Seçin
Tamamla

İlan Düzenleme ve Silme: Admin rolündeki kullanıcının açtığı ilanlar üzerinde değişiklik yapabildiği sistemdir.

Admin Paneli
Sistem Yönetimi
Yeni Başvuru
Açıklama
Başlangıç Tarihi
Bitti Tarihi
Genel Bilgi (Görünür)
Kullanıcı Seçin
Tamamla
İlan Ekleme
Yeni Başvuru
Açıklama
Başlangıç Tarihi
Bitti Tarihi
Genel Bilgi (Görünür)
Kullanıcı Seçin
Tamamla

Jüri Paneli:

Jüri Değerlendirme: Jürinin kendi atandığı ilanlara gelen başvuruları incelediği ve değerlendirdiği paneldir.

Aday Paneli:

Başvuru Ekranı: Başvuracak adayın, admin tarafından açılan ilanları gördüğü paneldir.

Başvuru PDF Formatı: Yapılan başvuru sonrasında adayın bilgilerinin PDF formatında tabloya dönüştüğü ve yaptığı çalışmalar, yazdığı makaleler neticesinde puan hesabının yapıldığı paneldir.

VII. SONUÇ

Bu proje kapsamında geliştirilen Akademik Personel Başvuru Sistemi, üniversitelerde yürütülen akademik atama süreçlerini dijitalleştirerek daha şeffaf, denetlenebilir ve kullanıcı dostu bir yapıya kavuşturmak amaçlanmıştır.

Rol tabanlı yapı sayesinde sistem; aday, jüri, yönetici ve admin gibi farklı kullanıcı profillerinin görev ve yetkilerini açık bir şekilde ayırmakta ve her birine özel arayüzler sunmaktadır.

Flutter kullanılarak geliştirilen ön yüz, hem mobil hem de web ortamlarında çalışabilirken; Node.js ve MySQL tabanlı arka uç yapısı ise güçlü bir veri yönetimi ve işlem performansı sağlamaktadır.

Giriş ve kayıt işlemlerinde TC Kimlik Numarası ile yapılan kimlik doğrulama ve SOAP servis entegrasyonu gibi adımlar, sistemin gerçek dünyadaki uygulamalarla entegre olabilecek düzeyde tasarlanmıştır. Sonuç olarak bu proje, teknik altyapı ve gerçekçi yapısı ile iyi bir tecrübe olmuştur.

VIII. KAYNAKÇA

- [1] [Dart ve Flutter paketleri için resmi depo.](#)
- [2] [Newest 'flutter' Questions - Stack Overflow](#)
- [3] [Flutter - Build apps for any screen](#)
- [4] [#1 || Bloc Introduction || Flutter Bloc Tutorial](#)
- [5] [01 Flutter API call | Flutter REST API Series](#)