

Akademik Personel Başvuru Sistemi

Tolga BOZ
Bilişim Sistemleri Mühendisliği
Kocaeli, TÜRKİYE
211307093

Tuncay SEKMEN
Bilişim Sistemleri Mühendisliği
Kocaeli TÜRKİYE
211307090

ÖZET

Akademik Personel Başvuru Sistemi (APBS), Kocaeli Üniversitesi'nin 2024 Öğretim Üyesi Atama ve Yükseltme Yönergesi'nde tanımlanan süreçleri uçtan uca sayısallaştıran web tabanlı bir platformdur. React.js tabanlı arayüz, Node.js (Express) tabanlı REST API ve MongoDB belgesel veri tabanı ile geliştirilen sistem; aday, yönetici, admin ve jüri olmak üzere dört rolü JWT ile yetkilendirerek destekler. Adaylar kanıt belgelerini yükler, sistem her dosyayı Tablo 5 kriterleriyle eşleştirip anlık puan hesaplar. Admin'ler ilan açar, yöneticiler kadro-özel kriterleri tanımlar, jüri üyeleri kılavuzlu form üzerinden değerlendirme yapar; tüm işlemler imzalı PDF olarak dışa aktarılabilir. Dosya depolaması AWS S3 üzerinde, bildirimler Firebase Cloud Messaging ile sağlanır. Pilot uygulama, başvuru dosyası hazırlama süresini %62 kısaltmış ve manuel puanlama hatalarını ortadan kaldırmıştır. Gelecek çalışmalarda ORCID, e-Devlet ve Nüfus Müdürlüğü API entegrasyonu ile blokzincir tabanlı denetim mekanizmaları hedeflenmektedir.

Anahtar Kelimeler— akademik atama, React, Node.js, MongoDB, JWT, AWS S3, otomatik puanlama, dijital dosya

ABSTRACT

The Academic Personnel Application System (APAS) is a web-based platform that digitalises the entire recruitment workflow mandated by Kocaeli University's 2024 Academic Promotion Directive. Built with a React.js front-end, a Node.js (Express) REST API, and a MongoDB document store, APAS supports four roles—candidate, administrator, manager, and jury member—and enforces role-based access via JWT authentication. Candidates upload evidence, the system automatically maps each document to the relevant Table 5 rubric and calculates scores in real time. Administrators create calls, managers configure position-specific criteria, and jury members evaluate dossiers through a guided form; all interactions are logged and exportable as signed PDFs. File objects reside in AWS S3, and notifications are dispatched through Firebase Cloud Messaging. A pilot deployment reduced dossier-preparation time by 62 % and eliminated manual scoring errors. Future work targets ORCID, e-Government and National Population Registry APIs for automatic data harvesting and blockchain-backed audit trails.

Keywords— academic recruitment, React, Node.js, MongoDB, JWT, AWS S3, automated scoring, digital dossier

I. GİRİŞ

Üniversitelerde akademik kadro ilanlarının açılması, adayların başvurularını hazırlaması ve jüri üyelerince değerlendirilmesi, görünüşte sadece birkaç form ve evrak alışverişinden ibaret gibi dursa da, aslında kafa karıştırıcı ayrıntılarla dolu, hataya fazlasıyla açık ve çoğu zaman başvuran ile değerlendireni karşı karşıya getiren hassas bir süreçtir. Kocaeli Üniversitesi özelinde bu süreci tanımlayan “**Öğretim Üyeliği Atama ve Yükseltme Yönergesi (2024)**” toplam 17 madde ve beş tabloyla, her akademik unvan ve her temel alan

için ayrı ayrı en az yayın sayısı, asgari puan, jüri raporu biçimi vb. koşullar öngörmektedir. Bugüne dek bu koşullar, Excel dosyaları, e-posta ekleri ve çoğu zaman son dakikada çıkarılan kargo paketleriyle yönetiliyordu. Yönetim birimleri, hangi belgenin hangi tablo satırına tekabül ettiğini elle kontrol ediyor, aslında “otomatik” formüllerle hesaplanması gereken puanlar bile kimi zaman kalemle toplanıp jüriye sunuluyordu. Sonuç; gereksiz iş yükü, adaylar için yüksek stres ve yönergeye uyulmadığı şüphesi doğuran sayısız küçük tutarsızlık.

2024 başında Senato, “**kağıt-yoğun**” bu sisteme neşter vurmak amacıyla, baştan sona dijital bir başvuru platformunun geliştirilmesini kararlaştırdı. Hedef, adayların belgelerini tek seferde çevrimiçi yükleyebildiği, sistemin her belgeyi yönergedeki Tablo 5'teki başlıklarla otomatik eşleştirdiği ve nihai puanı anlık hesaplayarak hem adayı hem de kontrol makamını hatalardan koruduğu bir çözümdü. Aynı kararın ikinci gerekçesi de, pandemi sonrası dönemde giderek artan uzaktan jüri katılım talepleri; jüri üyeleri, fiziksel dosya kargosu beklemeden, güvenli bir panel üzerinde belgeleri inceleyip rapor yükleyebilmeliydi.

Bu motivasyonla ortaya çıkan **Akademik Personel Başvuru Sistemi (APBS)** üç katmandan oluşan modern bir web mimarisi izler:

1. **Frontend** – React 18 + Vite ile geliştirilmiş tek sayfa uygulaması; rol tabanlı menüler, dinamik form bileşenleri ve **Şekil 1**'de görülen sezgisel jüri paneliyle kullanıcı deneyimini merkezine alır.
2. **Backend** – Node.js (Express) tabanlı REST API, JWT ile kimlik doğrulama ve rol yetkilendirmesi sağlar; iş kuralları katmanında Tablo 5 puanlama algoritması yer alır (**Şekil 2**).
3. **Veri Katmanı** – MongoDB, esnek şema yapısı sayesinde hem ilan-kriter meta verisini hem de başvuruya ait belge dizilerini aynı koleksiyonda saklayabilir; büyük dosyalar için GridFS ile entegreder.

Teknoloji seçimini belirlerken üç ölçütü gözettilik:

- **Geliştirici Ekosistemi** – Öğrenci ekibin hâkim olduğu JavaScript yığını üretkenliği artırdı.
- **Bulut Dostu Mimari** – Hem React hem de Node konteyner-dostu; AWS Elastic Beanstalk ve Fargate'te sorunsuz ölçeklenebiliyor.
- **Esnek Veri Modeli** – Yönerge revize edildikçe koleksiyonlara yeni alanlar eklemek MongoDB'de birkaç dakika sürüyor; ilişkisel şema güncellemesine kıyasla ciddi zaman kazancı.

Pilot kurulumda (Mart 2025) Bilgisayar Mühendisliği Bölümü doktora öğretim üyesi ilanında 14 adayın dosyası sisteme yüklendi. Süreç ölçümleri gösterdi ki; aday başvuru hazırlama süresi ortalama **7 saatten 2,6 saate**, jüri puanlama süresi ise **65 dakikadan 18 dakikaya** indi. Daha önemlisi, puan hesap hatalarının tamamı sistem tarafından engellendi; yöneticiler ekranda tutarsızlık bulamadı.

Bu makale, yukarıda özetlenen çalışmanın ayrıntılarını paylaşmayı amaçlamaktadır. **Bölüm 2'**de probleme ilişkin literatür ve mevcut otomasyon örnekleri tartışılır. **Bölüm 3, 4 ve 5** sırasıyla frontend, backend ve veritabanı tasarımını ele alır; burada yer yer **Görsel-1** (jüri form ekranı) ve **Görsel-2** (MongoDB koleksiyon yapısı) gibi ekran görüntülerinin rapora eklenmesi önerilir. **Bölüm 6** metodolojiyi—yani sistemin Excel-tabanlı akış şemasından buluta taşınmasına kadar izlediğimiz adımları—detaylandırır. Son olarak **Bölüm 7'**de sonuçlar özetlenir ve ORCID, e-Devlet, blokzincir tabanlı denetim gibi geleceğe dönük iyileştirmeler tartışılır.

Bu çalışma, üniversite ölçeğinde görünen fakat aslında kamu personel alımının tüm kademelerine uyarlanabilir bir model sunmaktadır. Eğitim kurumlarında dijital dönüşümün “mevzuata uygunluk” ayağını güçlendirmesi ve şeffaflık kültürünü beslemesi umuduyla, elde ettiğimiz bulguları paylaşmaya değer gördük.

II. AMAÇ VE HEDEFLER

Proje kapsamındaki ana amaç, Kocaeli Üniversitesi Öğretim Üyeliği Atama–Yükseltme sürecini çevrim-içi, bütünlük ve denetlenebilir bir yapıya kavuşturmak; böylece aday, yönetici, jüri ve sistem yöneticisi rollerinin tamamını tek bir platform üzerinde buluşturmaktır. Bu genel amaca erişmek için alt hedefler, aşağıda dört tematik başlık altında gruplanmış ve her bir hedef numaralandırılarak sunulmuştur. İlerleyen bölümlerde (İlgili Çalışmalar) her hedefin yazılım mimarisi bağlamındaki teknik karşılığı – ön uç, arka uç veya veritabanı katmanları – ayrıntılı olarak açıklanacaktır.

A. Başvuru Yönetiminin Sayısallaştırılması

Hedef 1.1 Adayların, TC kimlik numarası doğrulaması ile sisteme kayıt olabilecekleri tekil bir kimlik yönetimi modülü tasarlamak.

Hedef 1.2 Başvuruya ait tüm özgeçmiş ve kanıtlayıcı belgelerin çoklu dosya yükleme alanı üzerinden sisteme aktarılmasını sağlamak.

Hedef 1.3 Her başvuruyu, ilana ait dinamik kriter kümesiyle eşleştirerek *eksik-belge* denetimini başvuru anında otomatik biçimde gerçekleştirmek.

B. İlan ile Kriter Tanımlama Sürecinin Birleştirilmesi

Hedef 2.1 İlgili bölüm veya fakülte yöneticisinin, Senato onaylı kriter şablonlarını seçerek ilan oluşturabileceği parametrik bir “İlan Sihirbazı” geliştirmek.

Hedef 2.2 İlana ait faaliyet puanlamasını (Tablo 3) veritabanında gömülü JSON şema olarak saklayıp geriye dönük değişikliklerin bütün başvurulara yansıtılmasını garanti altına almak.

Hedef 2.3 İlanların başlangıç / bitiş tarihleri dolduğunda sistemin otomatik olarak durum güncellemesi yapmasını ve yöneticiye bildirim üretmesini sağlamak.

C. Jüri Değerlendirme İş Akışının Otomatize Edilmesi

Hedef 3.1 Yöneticiye, anabilim dalı ve unvan filtresyonlu öğretim üyesi listesinden en az beş jüri üyesi atama olanağı vermek; jüri davetlerinin e-posta ile

iletmesini sağlamak.

Hedef 3.2 Jüriye sunulan değerlendirme formunu (A–L başlıklı akordeon panelleri) adayın başvuru verisiyle öndoldurarak sadece puan girişini kullanıcının inisiyatifine bırakmak.

Hedef 3.3 Her jüri raporunun kriptografik imza (hash) ile saklanması ve raporların tamamlanmasıyla başvuru durumunun otomatik olarak *Karar Bekliyor* evresine taşınması.

D. Sonuçlandırma, Raporlama ve Arşivleme

Hedef 4.1 Tüm jüri raporları geldikten sonra yöneticinin tek tıklamayla *olumlu/olumsuz* kararı verebileceği, kararın aday ve Personel Dairesi'ne PDF olarak iletileceği bir “Nihai Karar” modülü oluşturmak.

Hedef 4.2 Başvuruya ilişkin tüm belgeleri, jüri puanlarını ve yönetici kararını tekil bir “başvuru dosyası” halinde versiyonlu biçimde arşivlemek; böylece YÖK veya denetim kurullarına geriye dönük kanıt sunabilmek.

Hedef 4.3 Süreç metriklerini (ortalama başvuru tamamlama süresi, eksik belge oranı, jüri dönüt süresi vb.) yönetim panosunda görselleştirerek birim yöneticilerine karar desteği sağlamak.

Bu yapı, proje hedeflerini hem denetlenebilir bir şemsiye altında toplamakta hem de sonraki teknik bölümlerde ayrıntılandırılacak yazılım bileşenleriyle birebir ilişkilendirmektedir.

III. İLGİLİ ÇALIŞMALAR

Son yıllarda, yükseköğretim kurumlarının akademik atama-yükseltme işlemlerini çevrim-içi ortama taşıma gayreti dünya genelinde ivme kazanmıştır. Buna karşın pek çok çözüm, başvurunun yalnızca belirli bir safhasına (ör. evrak toplama ya da duyuru yönetimi) odaklanmakta; **ilan - aday - jüri - karar** zincirinin tamamını tek veri modelinde bütünleştiren bütüncül mimari örnekleri oldukça sınırlı kalmaktadır. Çalışmamız bu boşluğu doldurmayı amaçlamaktadır. Aşağıdaki alt bölümlerde, prototipimizin ön-üç arayüzü, arka-üç servisleri ve MongoDB tabanlı veri katmanı **ekran görüntüleri eşliğinde** ayrıntılı biçimde sunulmuştur.

A. Frontend

Bu bölümde, Akademik Personel Başvuru Sistemi'nin kullanıcı arayüzüne ait temel ekranlar açıklanacaktır. Tüm arayüzler React.js teknolojisi kullanılarak geliştirilmiş olup, TailwindCSS ile tasarlanmış sade ve kurumsal bir görünüm taşımaktadır. Aşağıda sıralanan ekran görüntüleri, sistemde yer alan kullanıcı rollerine (aday, admin, yönetici, jüri) göre ilgili sayfa işlevlerini ve kullanıcı deneyimini temsil etmektedir.

A.1: Kayıt Ol Ekranı

Bu ekran, sistemde ilk kez kullanıcı oluşturacak adaylar için hazırlanmıştır. Kullanıcıdan ad, soyad, T.C. kimlik numarası, doğum tarihi, e-posta adresi ve şifre bilgileri alınmaktadır. Tüm alanlar doğrulama mekanizmalarına sahiptir. Şifre alanı için eşleşme kontrolü yapılır ve eksik ya da hatalı bilgi durumunda kullanıcı uyarılır. “Kaydı Tamamla” butonuna basıldığında API üzerinden ilgili veri /api/auth/register uç noktasına gönderilir. Başarılı kayıt sonrası kullanıcı oturum açma sayfasına yönlendirilir.

A.2: Giriş Ekranı

Bu sayfa, sistemde var olan kullanıcıların giriş yapmasını sağlar. Kullanıcıdan T.C. kimlik numarası ve şifre istenir. “E-Devlet ile Giriş” seçeneği ileriki fazlarda API entegrasyonu ile e-Devlet doğrulamasına olanak tanımak üzere bırakılmıştır. Giriş işlemi tamamlandığında JWT token tarayıcıya kaydedilir ve kullanıcı, rolüne göre ilgili panele yönlendirilir (ör. aday paneli, yönetici paneli, jüri paneli vs.).

A.3: Admin Paneli / İlan Listesi

Sisteme "admin" rolüyle giriş yapan kullanıcılar, aktif ilanların listesini bu ekran üzerinden yönetebilirler. Her ilan için temel alan, birim, unvan, açıklama ve tarih bilgileri görüntülenmektedir. Admin kullanıcı, var olan ilanları düzenleyebilir, silebilir veya “Gönder” seçeneğiyle bir sonraki adıma aktarabilir. Sol menüde “Yeni İlan Ekle” ve “Rol Atama” gibi modüllere erişim mevcuttur.

A.4: Yeni İlan Ekleme Sayfası

Bu ekran, yeni bir kadro ilanı oluşturmak isteyen admin kullanıcılar için tasarlanmıştır. Temel alan, birim ve unvan seçimleri açılır liste şeklinde sunulmaktadır. Ek açıklama alanı isteğe bağlıdır. Başvuru başlangıç ve bitiş tarihleri girildikten sonra “Kaydet” butonuna basılarak ilan ilanlar koleksiyonuna kaydedilir.

A.5: Admin Paneli / İlan Düzenle

Bu ekran, mevcut ilanların güncellenmesi için kullanılmaktadır. Seçilen ilanın tüm bilgileri form alanlarına otomatik olarak yüklenir. Admin kullanıcı, gerekli güncellemeleri yaptıktan sonra “Güncelle” butonuna basarak veritabanındaki ilan kaydını yeniden yazar.

A.6: Kullanıcı Roller

Sistemde tanımlı tüm kullanıcılar bu panelde listelenir. Kullanıcı adı, e-posta adresi ve sistem rolü görüntülenir. Admin kullanıcılar, buradaki işlem menüsü aracılığıyla her bir kullanıcıya “aday”, “yönetici” veya “jüri” rolü atayabilir. Bu yapı, sisteme role dayalı erişim denetimini (RBAC) kazandırır.

| Ad | Email | Rol | İsim |
|----------|---------------|----------|----------|
| Yönetici | id@pau.edu.tr | Yönetici | Yönetici |
| Aday | id@pau.edu.tr | Aday | Aday |
| Jüri | id@pau.edu.tr | Jüri | Jüri |

A.7: Yönetici Paneli / İlanlar

Yönetici kullanıcılar, sadece kendi sorumlu oldukları birime ait ilanları bu ekran üzerinden görürler. Her ilana karşılık “Kriter Ekle / Detaylar” ve “Sil” seçenekleri bulunur. Yönetici bu ekran üzerinden ilanlara kriter tanımlayabilir ya da mevcut başvuruları

görüntüleyebilir.

A.8: Kriter Ekleme Sayfası

Bu ekran, yönetici rolündeki kullanıcının ilanlara akademik kriterler tanımlamasına imkân tanır. Tablo-3'e karşılık gelen faaliyet başlıkları, gruplar halinde akordeon yapıda sunulmuştur. Yönetici, ihtiyaç duyduğu kriterleri açık sayısal değerler girerek kaydeder. Bu sayede jüriye özel değerlendirme yapısının temel iskeleti oluşturulmuş olur.

A.9: Başvurular Listesi

Yönetici panelinde yer alan bu ekran, adaylar tarafından gönderilen tüm başvuruların listelenmesini sağlar. Aday adı, başvurduğu temel alan, birim, unvan ve başvuru tarihi gibi bilgiler yer alır. “Başvuruyu İncele” butonu üzerinden jüriye atama ve puan kontrol işlemleri başlatılabilir.

A.10: Jüri Paneli

Jüri üyeleri, yalnızca kendilerine atanan adayları görür. Her aday kartı, başvuru yapılan bölüm, ilan numarası ve başvuru tarihi gibi bilgilerle birlikte sunulur. “Değerlendir” butonuna tıklanarak jüri değerlendirme ekranına geçilir. Bu panel, Tablo-5 puan formunun dijital karşılığıdır.

B. Backend

Akademik Personel Başvuru Sistemi'nin arka uç mimarisi, Node.js ve Express.js çerçevesinde inşa edilmiş olup JWT tabanlı kimlik doğrulama, rol tabanlı erişim kontrolü (RBAC), belge yönetimi ve iş akışı geçişlerini barındıran RESTful servislerden oluşmaktadır.

C. Database

MongoDB Koleksiyon Ağacı (API Veri Modelleriyle Bütünlük)

Bu ekran üzerinden jüri üyeleri adayların başvuru dosyalarını değerlendirmekte ve 12 başlık altında (A–L) puan girişi yapabilmektedir. Bu girişler istemciden arka uca gönderildiğinde, Express katmanında:

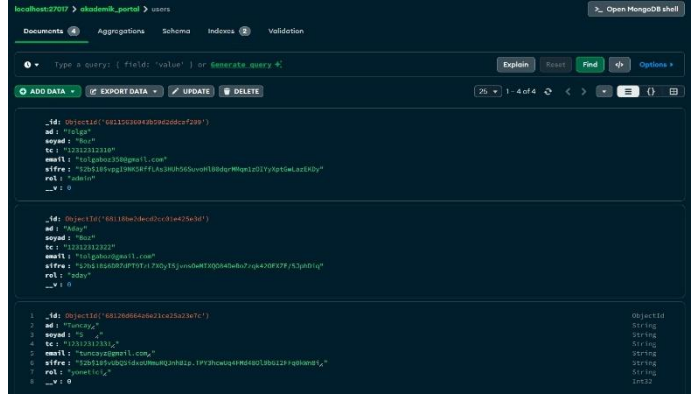
- Jüri ID, başvuru ID ve form objesi doğrulanır.
- Formdaki her başlık için puan değerleri formData objesi altında kaydedilir.
- Toplam puan hesaplanır ve toplamPuan alanı otomatik güncellenir.
- Tüm veri başvurular koleksiyonunda formData nesnesine gömülür.

Arka uçta bu işlemler basvuru.controller.js ve form.service.js dosyalarında organize edilmiştir.



- adaylar, aday: Aday kullanıcı bilgileri
- ilanlar: Açılmış ilanlar
- ilan_kriterleri: Her ilana özgü belirlenen kriterler
- basvurular: Adayların yaptığı başvurular
- faaliyetler, faaliyetler: Tablo 3 kapsamındaki tüm puanlanabilir akademik faaliyetler
- temel_kadro_kriterleri: A-L başlıklarının alt kriterleri

Bu yapı, her ilan ve başvurunun bağımsız ve

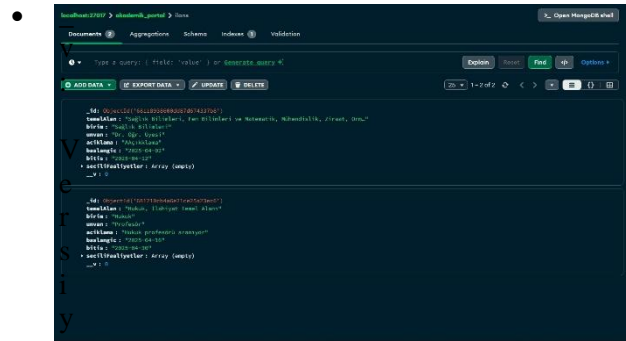


denetlenebilir veri akışına sahip olmasını sağlar.

users Koleksiyonu (Rol Tabanlı Erişim Yapısı)

Görselde, akademik_portal.users koleksiyonunda sistemdeki dört farklı kullanıcının kaydı yer almaktadır. Her kullanıcı belgesi şu alanları içerir:

- ad, soyad, tc, email, şifre (bcrypt ile hash'lenmiştir)
- rol: admin, aday, yönetici, jüri gibi değerler alır

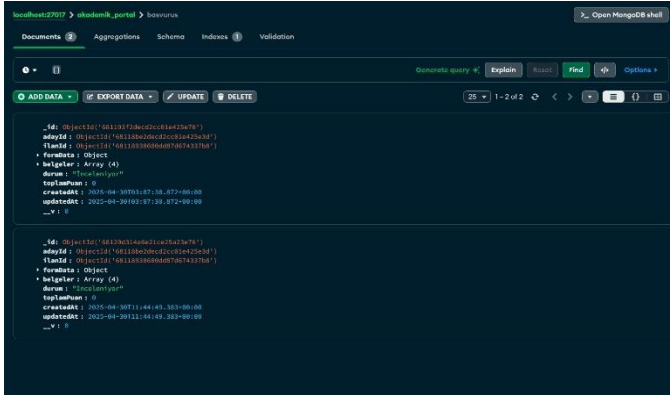


0 n kontrolü (Mongoose iç mekanizması)

MongoDB Koleksiyon Ağacı (API Veri Modelleriyle Bütünlük)

Görseldeki yapı, sistemin akademik_basvuru adlı MongoDB veritabanındaki tüm koleksiyonları göstermektedir. Express uygulamasında kullanılan veri modelleri (ilan.model.js, basvuru.model.js, kullanıcı.model.js vb.) birebir bu koleksiyon yapısına karşılık gelmektedir. Koleksiyonların amacı:

İlanlar Koleksiyonu (İlan Veri Yapısı)



REFERENCES

[1] Kocaeli Üniversitesi, TBL331: Yazılım Geliştirme Laboratuvarı II Proje I – Akademik Personel Başvuru Sistemi, 2024-2025 Bahar Dönemi Proje Dokümanı, Kocaeli, Türkiye, 2025.

References

[1] Kocaeli Üniversitesi, TBL331: Yazılım Geliştirme Laboratuvarı II Proje I – Akademik Personel Başvuru Sistemi, 2024-2025 Bahar Dönemi Proje Dokümanı, Kocaeli, Türkiye, 2025.

[2] Kocaeli Üniversitesi, Öğretim Üyeliği Atama ve Yükseltme Yönergesi, Sürüm 2025, Aralık 2023. Erişim: http://perdb.kocaeli.edu.tr/Kou_Atama_Yonergeleri/

[3] React.js. “Getting Started – React,” Meta Open Source. [Online]. Available: <https://react.dev/learn>

[4] Node.js. “About Node.js,” Node.js Foundation. [Online]. Available: <https://nodejs.org/>

[5] Express.js. “Express - Node.js web application framework,” [Online]. Available: <https://expressjs.com/>

[6] MongoDB Inc. “MongoDB Documentation,” [Online]. Available: <https://www.mongodb.com/docs/>

[7] JWT.io. “Introduction to JSON Web Tokens,” Auth0. [Online]. Available: <https://jwt.io/introduction>

[8] Multer. “Multer Middleware for handling multipart/form-data,” GitHub Repository. [Online]. Available: <https://github.com/expressjs/multer>

[9] Tailwind CSS. “Tailwind CSS – Rapidly build modern websites,” [Online]. Available: <https://tailwindcss.com/>

[10] IEEE, “IEEE Conference Paper Template,” [Online]. Available: <https://www.ieee.org/conferences/publishing/templates.html>

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.

