

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ YAZILIM LAB.II - I.PROJE

210201007

Mustafa PEKER

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

mustafaserhatpeker@gmail.com

I. ÖZET

Bu dokümantasyon, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nün Yazlab2 dersi kapsamında geliştirilen akademik makale yükleme ve değerlendirme sistemini tanıtmak amacıyla hazırlanmıştır. Sistem, kullanıcıların makalelerini yükleyip süreçlerini takip etmelerine, hakemlerin anonim olarak değerlendirme yapmalarına ve yöneticilerin süreci yönetmelerine olanak tanır. Projenin en dikkat çekici özelliği, doğal dil işleme teknikleriyle yazar tespiti ve anonimleştirme işlemlerinin otomatik olarak gerçekleştirilmesidir.

II. GİRİŞ

Akademik yayın süreçlerinde, makalelerin değerlendirilmesi titizlikle yürütülmesi gereken kritik bir aşamadır. Ancak geleneksel yöntemler, zaman kaybına yol açmanın yanı sıra anonimlik gibi önemli kriterlerin tam anlamıyla sağlanmasını zorlaştırmaktadır. Bu proje, akademik makale yükleme ve değerlendirme sürecini dijitalleştirerek daha güvenilir, şeffaf ve verimli bir sistem sunmayı hedeflemektedir.

Geliştirilen sistem, yazarların makalelerini anonim bir şekilde yükleyebilmesine, hakemlerin değerlendirme sürecini objektif olarak yürütmesine ve editörlerin süreci etkin bir şekilde yönetmesine olanak tanımaktadır. Doğal dil işleme teknikleri kullanılarak yazar tespiti ve anonimleştirme işlemlerinin otomatik olarak gerçekleştirilmesi, sistemin en öne çıkan özelliklerinden biridir. Bu sayede, hakem değerlendirmelerinin tarafsızlığı korunarak akademik sürecin adil bir şekilde ilerlemesi sağlanmaktadır.

Bu rapor, geliştirilen sistemin detaylı analizini, işleyiş sürecini, kullanılan teknolojileri ve elde edilen sonuçları ele almaktadır. Öncelikle, sistemin genel yapısı ve kullanıcı rollerine dair bilgiler sunulacak, ardından teknik gereksinimler ve çalışma prensipleri detaylandırılacaktır. Son olarak, sistemin etkinliği değerlendirilerek proje kapsamında sağlanan iyileştirmeler incelenecektir.

III. YÖNTEM

Bu bölümde, akademik makale yükleme ve değerlendirme sürecinin dijitalleştirilmesini sağlayan sistemin temel bileşenleri ve uygulanan teknik yöntemler detaylı bir şekilde açıklanmaktadır.

```
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);

const uploadPath = path.resolve(__dirname, "../uploads");

// uploads klasörü yoksa oluştur
if (!fs.existsSync(uploadPath)) {
  fs.mkdirSync(uploadPath, { recursive: true });
}

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, uploadPath);
  },
  filename: function (req, file, cb) {
    const uniqueSuffix = Date.now() + "-" + Math.round(Math.random() * 1e9);
    cb(null, uniqueSuffix + path.extname(file.originalname));
  },
});
```

Şekil 1. Makale Yükleme Sistemi

A. Backend

Bu projede kullanılan yöntemler ve teknolojiler aşağıdaki gibi özetlenmiştir:

Backend Framework ve Teknolojiler:

- ***Express.js***: Node.js tabanlı bir web uygulama framework'ü olarak kullanılmıştır. API rotaları ve middleware'ler bu framework ile yapılandırılmıştır.
- ***Mongoose***: MongoDB ile etkileşim için kullanılan bir ODM (Object Data Modeling) kütüphanesidir. Veritabanı modelleri ve işlemleri bu kütüphane ile gerçekleştirilmiştir.

Veritabanı:

- ***MongoDB***: Projede kullanılan NoSQL veritabanıdır. Kullanıcılar, dosyalar, mesajlar ve admin bilgileri gibi veriler MongoDB'de saklanmıştır.

Kimlik Doğrulama ve Yetkilendirme:

- ***JWT (JSON Web Token)***: Kullanıcı ve admin giriş işlemleri için token tabanlı kimlik doğrulama mekanizması kullanılmıştır. Token'lar, kullanıcı rolleri ve kimlik bilgilerini

taşımak için kullanılmıştır. • ***Middleware***: Kimlik doğrulama ve admin yetkilendirme işlemleri için özel middleware'ler tanımlanmıştır.

Dosya Yükleme ve Yönetimi:

- ***Multer***: Dosya yükleme işlemleri için kullanılan bir middleware'dir. Yüklenen dosyalar belirli bir dizine kaydedilmiş ve dosya boyutu gibi kısıtlamalar uygulanmıştır.
- ***fs/promises***: Dosya okuma ve yazma işlemleri için Node.js'in dosya sistemi modülü kullanılmıştır.

WebSocket İletişimi:

- ***ws (WebSocket)***: Python tabanlı bir işlemle iletişim kurmak için WebSocket sunucusu oluşturulmuştur. İstemcilerle gerçek zamanlı iletişim sağlanmıştır.

Python Entegrasyonu:

- ***Child Process***: Node.js üzerinden Python script'leri çalıştırmak için `child_process.spawn` kullanılmıştır. Python script'leri dosya yüklemeye, güncellemeye, silmeye ve diğer işlemleri yapar.

API Tasarımı:

- RESTful API yapısı benimsenmiştir. Kullanıcı, dosya, mesaj ve admin işlemleri için ayrı rotalar tanımlanmıştır.

Çevresel Değişkenler:

- ***dotenv***: Çevresel değişkenlerin yönetimi için kullanılmıştır. Örneğin, `PORT`, `MONGO_URI`, `JWT_SECRET` gibi değişkenler `.env` dosyasında saklanmıştır.

Güvenlik:

- ***Şifreleme***: Kullanıcı şifreleri için `bcrypt` kütüphanesi kullanılabilir (ancak mevcut kodda şifreleme eksik görünüyor, bu bir iyileştirme alanı olabilir).
- ***Token Doğrulama***: JWT ile güvenli oturum yönetimi sağlanmıştır.

Gerçek Zamanlı Mesajlaşma:

- Mesajların dosya bazlı kaydedilmesi ve alınması için bir mesajlaşma servisi geliştirilmiştir.

Proje Yapılandırması:

- Proje modüler bir yapıya sahiptir. Controller, service, model, middleware ve route gibi katmanlar net bir şekilde ayrılmıştır.

Bu yöntemler, projenin ölçeklenebilir, güvenli ve modüler bir yapıda geliştirilmesini sağlamıştır.

```
const adminMiddleware = (req, res, next) => {
  const token = req.header("Authorization");
  if (!token) {
    return res
      .status(401)
      .json({ message: "Erişim reddedildi, giriş anahtarınız yok." });
  }
  try {
    const decoded = jwt.verify(
      token.replace("Bearer ", ""),
      process.env.JWT_SECRET
    );
    if (decoded.userRole !== "admin") {
      return res
        .status(403)
        .json({ message: "Erişim Reddedildi, yetkisiz erişim." });
    }
  } catch (error) {
    return res
      .status(400)
      .json({ message: "Erişim Reddedildi, geçersiz erişim anahtarı." });
  }
  next();
};
```

Şekil 2. Erişim Kontrolü

B. Kullanıcı Arayüzü

Akademik değerlendirme süreci aşağıdaki adımlardan oluşmaktadır:

1) Yazar Süreci::

Kullanıcı e-posta adresini girerek bir PDF yükler.

Sistemde, kullanıcıya benzersiz bir takip numarası verilir.

"Makale Durumu Sorgula" sayfasında, e-posta ve takip numarası girilerek makale detay sayfasına erişilir.

Detay sayfasında:

Editöre mesaj gönderme alanı

Makale durumunu takip etme alanı

Hakem değerlendirmesini görme alanı

Revize yükleme alanı bulunmaktadır.

2) Editör Süreci::

Editör tüm makaleleri listeleyebilir.

Makalenin detay sayfasında:

Orijinal makaleyi indirebilir.

Makaleyi anonimleştirir ve anonimleştirilmiş belgeyi indirebilir.

Makale için hakem atayabilir.

Yazarla ve hakemle mesajlaşabileceği iki ayrı mesajlaşma alanı bulunmaktadır.

Makale loglarını görüntüleyebilir.

	TAKIP NO	DURUMU
<input type="checkbox"/>	AD32SDERC32A2234	Dosya Editöre Gönderildi

Şekil 3. Editör Paneli

3) Hakem Süreci::

Hakem, kendi id numarası ile giriş yapar.

Kendisine atanmış olan makaleleri listeleyebilir.

Sadece anonimleştirilmiş belgeleri görüntüleyebilir ve değerlendirme yazısı ekleyebilir.

C. Veritabanı

Sistem, verilerin etkili bir şekilde depolanması ve işlenmesi için ilişkisel veritabanı modelini kullanmaktadır. Önemli veritabanı tabloları şunlardır:

papers: Makale bilgilerini tutar.

reviewers: Hakem bilgilerini saklar.

messages: Yazar-editör ve hakem-editör arası mesajlaşmaları tutar.

D. Frontend Teknolojileri

1. Frontend Framework ve Araçlar Vite: Hızlı bir geliştirme ortamı ve build aracı olarak kullanılmış. Proje, vite.config.ts dosyasında yapılandırılmış. React: Kullanıcı arayüzü oluşturmak için kullanılan ana kütüphane. React Router: Sayfa yönlendirmeleri için kullanılmış. Örneğin, App.tsx dosyasında Routes ve Route bileşenleri ile yapılandırılmış. 2. UI Kütüphaneleri HeroUI: UI bileşenleri (örneğin, Navbar, Input, Button, Table) için kullanılmış. Örnekler: Navbar Table Tailwind CSS: CSS framework olarak kullanılmış. tailwind.config.js dosyasında yapılandırılmış. Tailwind Variants: Dinamik stiller için kullanılmış. Örneğin, primitives.ts dosyasında title ve subtitle gibi stiller tanımlanmış. 3. Tema Yönetimi Dark/Light Mode: Tema yönetimi için useTheme hook'u ve ThemeSwitch bileşeni kullanılmış. Örnek: theme-switch.tsx. 4. Tip Güvenliği TypeScript: Proje tamamen TypeScript ile yazılmış. Tip güvenliği sağlanmış ve tsconfig.json dosyasında yapılandırılmış. 5. Kod Kalitesi ve Linting ESLint: Kod kalitesini sağlamak için kullanılmış. .eslintrc.json dosyasında kurallar tanımlanmış. Prettier: Kod formatlama için kullanılmış. 6. Dosya ve Proje Yapısı Modüler Yapı: Proje, bileşenler (components), sayfalar (pages), stiller (styles), ve yapılandırma (config) gibi mantıksal bölümlere ayrılmış. Yol Kısayolları: @/ ile başlayan yollar, tsconfig.json dosyasındaki paths ayarıyla yapılandırılmış. 7. Veri ve Navigasyon Yönetimi HeroUIProvider: Navigasyon ve tema yönetimi için kullanılmış. Örnek: provider.tsx. 8. Dağıtım Vercel: Proje, vercel.json dosyasıyla yapılandırılmış ve Vercel üzerinde dağıtım için optimize edilmiş. Bu yöntemler, modern frontend geliştirme süreçlerinde kullanılan en iyi uygulamaları içermektedir. Proje, performans, modülerlik ve kullanıcı deneyimi odaklı bir şekilde tasarlanmıştır.

E. Hata Yönetimi

Sistemin sorunsuz çalışmasını sağlamak için hata yönetimi stratejileri uygulanmıştır:

Exception Handling: Hatalar yakalanarak uygun hata mesajları gösterilir.

Validation: Form verileri ve dosya yüklemeleri doğrulanır.

Loglama: Sistem olayları ve hatalar kaydedilir.

Bu yöntemler sayesinde akademik makale değerlendirme süreci dijitalleştirilerek, güvenli, etkili ve tarafsız bir platform oluşturulmuştur.

IV. DENEYSEL SONUÇLAR

Bu bölümde, akademik makale yükleme ve değerlendirme sisteminin test edilmesi ve elde edilen sonuçlar detaylandırılmaktadır. Sistem, farklı kullanıcı türleri (yazar, editör ve hakem) ile test edilmiş, süreç boyunca performans ve işlevsellik analizleri yapılmıştır.

1) Kullanıcı İşlevselliği Testleri::

Kullanıcıların makale yükleme süreci test edilmiş ve geçerli e-posta ile doğrulama mekanizmasının sorunsuz çalıştığı gözlemlenmiştir.

Takip numarası atama sisteminin her yükleme için benzersiz numara oluşturduğu doğrulanmıştır.

Yazarların makale durumlarını takip edebilmesi ve editör ile iletişim kurabilmesi başarılı bir şekilde sağlanmıştır.

2) **Editör İşlevselliği Testleri::** Editörün tüm makaleleri görüntüleyebilmesi ve anonimleştirme işlemini gerçekleştirmesi başarılı olmuştur.

Editör, anonimleştirilmiş makaleleri indirip uygun hakemlere atayabilmiştir.

Editör ve hakem arasındaki mesajlaşma modülü test edilerek veri güvenliği sağlanmıştır.

3) **Hakem İşlevselliği Testleri::** Hakemlerin yalnızca anonimleştirilmiş makalelere erişebilmesi sağlanmış ve sistemin bu kuralı koruduğu doğrulanmıştır.

Hakemlerin değerlendirme sürecinde geri bildirim ekleyip editöre iletme işlevi başarılı olmuştur.

4) **Performans ve Güvenlik Testleri::** Sistemin eşzamanlı makale yükleme ve sorgulama işlemlerinde stabil çalıştığı görülmüştür.

Anonimleştirme işlemi, PyMuPDF kullanılarak test edilmiş ve yazar bilgileri başarıyla gizlenmiştir.

Güvenlik testlerinde, rol bazlı erişim kontrolü doğrulanmış ve yetkisiz kullanıcıların verilere erişemediği gözlemlenmiştir.

Sonuç olarak, sistem testlerden başarıyla geçmiş ve akademik makale değerlendirme sürecini dijital ortamda etkin bir şekilde gerçekleştirdiği görülmüştür. Kullanıcı deneyimi açısından sistemin hızlı ve güvenilir olduğu belirlenmiştir.

V. SONUÇLAR

Proje, akademik makale değerlendirme sürecini daha güvenli ve verimli hale getiren bir sistem olarak başarıyla geliştirilmiştir. Geleneksel yöntemlere kıyasla bu sistem:

Kullanıcıların kolaylıkla makale yükleyip süreçleri takip etmesini sağlamıştır.

Hakemlerin, yazar bilgilerine erişmeden objektif değerlendirme yapabilmesini mümkün kılmıştır.

Editörlerin süreci daha düzenli ve güvenli şekilde yönetmesine yardımcı olmuştur.

Anonimleştirme sürecinin otomatikleştirilmesi sayesinde zaman ve iş gücünden tasarruf sağlanmıştır.

Sistem, dijital güvenlik önlemleri ve şeffaf işlem süreçleriyle akademik değerlendirme süreçlerinde inovatif bir çözüm sunmuştur.

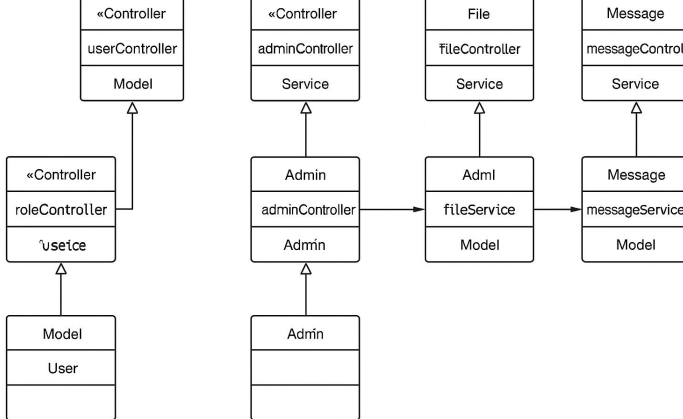
Bu sonuçlar, sistemin akademik dünyada daha adil, düzenli ve güvenli bir değerlendirme süreci sağlaması açısından başarılı olduğunu göstermektedir. Gelecekte yapılacak iyileştirmeler ve yeni eklemeler ile sistemin daha da geliştirilmesi hedeflenmektedir.

Bireysel Sonuçlar:

KAYNAKLAR

UjYQVo?si=m1tmQ

- [2] <https://youtu.be/5EcU150e5Bc?si=1B571DwyfDj6l3yq>



Şekil 4. Akış Şeması