



Kütüphane Yönetim Sistemi

Proje Teknik Dokümantasyonu

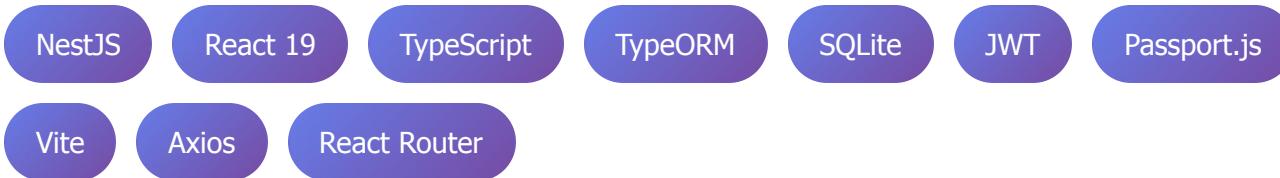
🔗 Proje Paylaşım Linki

GitHub Repository: <https://github.com/tercanomeer/web>

1. Proje Genel Bakış

Bu proje, modern web teknolojileri kullanılarak geliştirilmiş kapsamlı bir **Kütüphane Yönetim Sistemi**'dir. Sistem, kullanıcıların kitapları, yazarları ve kategorileri yönetmesine olanak tanır. Ayrıca kitap ödünç alma işlemlerini takip eder ve rol tabanlı yetkilendirme sistemi ile farklı kullanıcı tiplerinin farklı işlevlere erişimini kontrol eder.

Kullanılan Teknolojiler



Temel Özellikler

🔒 Kimlik Doğrulama

JWT tabanlı güvenli giriş/kayıt sistemi

👤 Rol Yönetimi

Admin, User ve Guest rolleri

📚 Kitap Yönetimi

💳 Ödünç Sistemi

Tam CRUD işlemleri ve resim yükleme

Kitap ödünç alma ve iade takibi

2. Backend API Endpoint'leri

Backend, NestJS framework'ü kullanılarak geliştirilmiştir. Tüm endpoint'ler (kimlik doğrulama hariç) JWT token ile korunmaktadır. Backend `http://localhost:3000` adresinde çalışmaktadır.

2.1 Kimlik Doğrulama (Auth) Endpoint'leri

Metod	Endpoint	Açıklama	Kullanım
<code>POST</code>	<code>/auth/register</code>	Yeni kullanıcı kaydı	email, password, name ve opsiyonel role parametreleri ile yeni kullanıcı oluşturur. Şifre otomatik olarak bcrypt ile hashlenir.
<code>POST</code>	<code>/auth/login</code>	Kullanıcı girişи	email ve password ile giriş yapar. Başarılı girişte JWT access_token ve kullanıcı bilgilerini döndürür.

2.2 Kitaplar (Books) Endpoint'leri

Kitap işlemleri için tüm endpoint'ler JWT korumalıdır.

Metod	Endpoint	Açıklama	Kullanım
<code>GET</code>	<code>/books</code>	Tüm kitapları listele	Sistemdeki tüm kitapları yazar ve kategori bilgileriyle birlikte döndürür.
<code>GET</code>	<code>/books/:id</code>	Belirli kitabı getir	ID'ye göre tek bir kitabın detaylı bilgilerini döndürür.
<code>POST</code>	<code>/books</code>	Yeni kitap oluştur	title, isbn, description, quantity, authorId ve categoryIds parametreleri ile yeni kitap ekler.

POST	/books/upload	Kitap resmi yükle	Multipart form-data ile kitap kapak resmi yükler. Maksimum 5MB, sadece jpg/jpeg/png/gif formatları kabul edilir.
PATCH	/books/:id	Kitap güncelle	Mevcut kitap bilgilerini günceller. Kısmi güncelleme desteklenir.
DELETE	/books/:id	Kitap sil	Kitabı ve varsa ilişkili resim dosyasını sistemden kaldırır.

2.3 Yazarlar (Authors) Endpoint'leri

Metod	Endpoint	Açıklama	Kullanım
GET	/authors	Tüm yazarları listele	Sistemdeki tüm yazarları döndürür.
GET	/authors/:id	Belirli yazarı getir	ID'ye göre tek bir yazarın bilgilerini döndürür.
POST	/authors	Yeni yazar oluştur	name ve bio parametreleri ile yeni yazar ekler.
PATCH	/authors/:id	Yazar güncelle	Mevcut yazar bilgilerini günceller.
DELETE	/authors/:id	Yazar sil	Yazarı sistemden kaldırır.

2.4 Kategoriler (Categories) Endpoint'leri

Metod	Endpoint	Açıklama	Kullanım
GET	/categories	Tüm kategorileri listele	Sistemdeki tüm kategorileri döndürür.
GET	/categories/:id	Belirli kategoriyi getir	ID'ye göre tek bir kategorinin bilgilerini döndürür.

POST	/categories	Yeni kategori oluştur	name ve description parametreleri ile yeni kategori ekler.
PATCH	/categories/:id	Kategori güncelle	Mevcut kategori bilgilerini günceller.
DELETE	/categories/:id	Kategori sil	Kategoriyi sistemden kaldırır.

2.5 Ödünç İşlemleri (Borrows) Endpoint'leri

Metod	Endpoint	Açıklama	Kullanım
GET	/borrows	Tüm ödünç kayıtlarını listele	Opsiyonel userId veya bookId query parametresi ile filtreleme yapılabilir.
GET	/borrows/:id	Belirli ödünç kaydını getir	ID'ye göre tek bir ödünç kaydının detaylarını döndürür.
POST	/borrows	Yeni ödünç kaydı oluştur	userId ve bookId parametreleri ile kitap ödünç alma işlemi başlatır.
PATCH	/borrows/:id/return	Kitap iade et	Ödünç alınan kitabı iade eder ve returnDate alanını günceller.
PATCH	/borrows/:id	Ödünç kaydı güncelle	Ödünç kayıt bilgilerini günceller.
DELETE	/borrows/:id	Ödünç kaydı sil	Ödünç kaydını sistemden kaldırır.

2.6 Kullanıcılar (Users) Endpoint'leri

Not: Bu endpoint'lere sadece ADMIN rolündeki kullanıcılar erişebilir.

Metod	Endpoint	Açıklama	Kullanım
GET	/users	Tüm kullanıcıları listele	Sistemdeki tüm kullanıcıları döndürür (sadece admin).

GET	/users/:id	Belirli kullanıcıyı getir	ID'ye göre tek bir kullanıcının bilgilerini döndürür.
POST	/users	Yeni kullanıcı oluştur	Admin tarafından yeni kullanıcı ekler.
PATCH	/users/:id	Kullanıcı güncelle	Kullanıcı bilgilerini veya rolünü günceller.
DELETE	/users/:id	Kullanıcı sil	Kullanıcıyı sistemden kaldırır.

3. Frontend Component'leri

Frontend, React 19 ve TypeScript kullanılarak geliştirilmiştir. Vite ile derlenmekte ve <http://localhost:3001> adresinde çalışmaktadır.

3.1 Ana Bileşenler (Components)

App

Uygulamanın ana bileşenidir. BrowserRouter ile yönlendirme yapısını kurar, AuthProvider ile kimlik doğrulama context'ini sağlar ve tüm sayfa rotalarını tanımlar. Kimlik durumuna göre sayfalara erişimi kontrol eder.

Layout

Tüm korumalı sayfalarda kullanılan ortak sayfa düzeni bileşenidir. Üst navigasyon çubuğunu içerir ve kullanıcı adı, rol bilgisi ile çıkış butonunu gösterir. Admin kullanıcıları için ek menü öğeleri sunar.

ProtectedRoute

Sayfa erişim kontrolü sağlayan bileşendir. Kimlik doğrulanmamış kullanıcıları giriş sayfasına, yetkisiz kullanıcıları ana sayfaya yönlendirir. Opsiyonel olarak rol bazlı kısıtlama uygular.

GuestRestrictionModal

Misafir (Guest) kullanıcıların yetkisiz işlem yapmaya çalışığında gösterilen uyarı modalıdır. CRUD işlemleri için User veya Admin rolü gerektiğini bildirir.

AuthContext

Uygulama genelinde kimlik doğrulama durumunu yöneten React Context'tir. login, register, logout fonksiyonlarını ve kullanıcı bilgilerini tüm bileşenlere sağlar.

3.2 Sayfa Bileşenleri (Pages)

Login

Kullanıcı giriş sayfasıdır. Email ve şifre ile giriş formu içerir. Başarılı girişte ana sayfaya yönlendirir, hata durumunda mesaj gösterir. Kayıt sayfasına link içerir.

Register

Yeni kullanıcı kayıt sayfasıdır. İsim, email, şifre alanları ve rol seçimi içerir. User, Admin veya Guest rollerinden biri seçilebilir. Başarılı kayıtta otomatik giriş yapar.

Dashboard

Ana sayfa / kontrol panelidir. Kullanıcıyı hoşgeldin mesajıyla karşılar ve Books, Authors, Categories sayfalarına hızlı erişim kartları sunar.

Books

Kitap yönetim sayfasıdır. Tüm kitapları listeler, yeni kitap ekleme, düzenleme ve silme işlemlerini sağlar. Kitap kapak resmi yükleme, yazar ve kategori seçimi özellikleri içerir. Ödünç alma/iade butonları bulunur.

Authors

Yazar yönetim sayfasıdır. Yazarları listeler ve CRUD işlemlerini sağlar. Her yazar için isim ve biyografi bilgisi yönetilir.

Categories

Kategori yönetim sayfasıdır. Kitap kategorilerini listeler ve yönetir. Her kategori için ad ve açıklama bilgisi içerir.

Users (Sadece Admin)

Kullanıcı yönetim sayfasıdır. Yalnızca admin rolündeki kullanıcılar görünür. Sistemdeki tüm kullanıcıları listeler, yeni kullanıcı ekler ve rol değişikliği yapar.

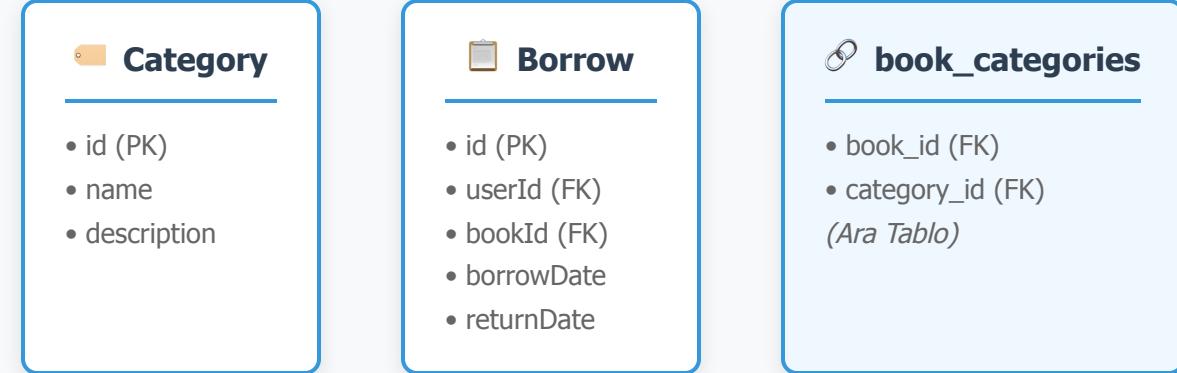
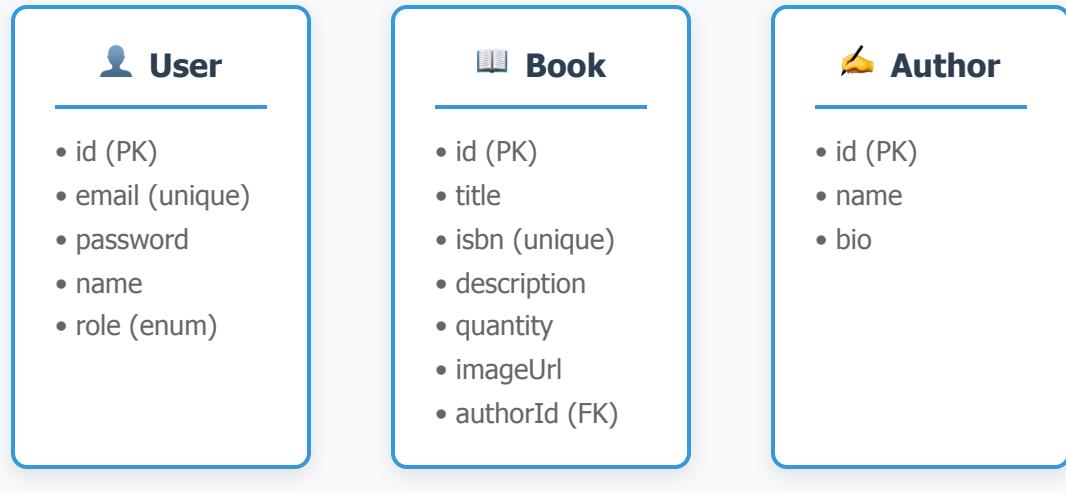
Borrows

Ödünç işlemleri takip sayfasıdır. Aktif ve tamamlanmış ödünç kayıtlarını listeler. Kitap iade ve kayıt silme işlemlerini sağlar. Admin kullanıcıları tüm kayıtları, diğer kullanıcılar sadece kendi kayıtlarını görebilir.

4. Veritabanı Diyagramı

Proje SQLite veritabanı kullanmaktadır. TypeORM ile entity'ler tanımlanmış ve ilişkiler kurulmuştur.

Entity-Relationship Diyagramı



İlişkiler:

- **Author → Books** (1:N) — Bir yazar birden fazla kitap yazabilir
- **Book ↔ Category** (M:N) — Bir kitap birden fazla kategoriye ait olabilir
- **User → Borrows** (1:N) — Bir kullanıcı birden fazla kitap ödünç alabilir
- **Book → Borrows** (1:N) — Bir kitap birden fazla kez ödünç alınabilir

5. Kullanıcı Rollerleri

Rol	Yetkiler
Admin	Tüm CRUD işlemleri, kullanıcı yönetimi, tüm ödünç kayıtlarını görme
User	Kitap, yazar, kategori CRUD işlemleri, kitap ödünç alma/iade, kendi ödünç kayıtlarını görme
Guest	Sadece görüntüleme (read-only), CRUD işlemleri yapamaz

6. Kurulum ve Çalıştırma

Gereksinimler

- Node.js (v18 veya üzeri)
- npm veya yarn

Backend

1. cd backend — Backend dizinine git
2. npm install — Bağımlılıkları yükle
3. npm run start:dev — Geliştirme modunda başlat

Backend <http://localhost:3000> adresinde çalışacaktır.

Frontend

1. cd frontend — Frontend dizinine git
2. npm install — Bağımlılıkları yükle
3. npm run dev — Geliştirme modunda başlat

Frontend <http://localhost:3001> adresinde çalışacaktır.

Ömer Tercan - 22253502

Kütüphane Yönetim Sistemi © 2026