

# Kütüphane Otomasyon Sistemi

Emre Yüksel  
221307103  
emreyuksell78@gmail.com

Egehan Sözen  
221307013  
egehansozenn@gmail.com

Yunus Emir Atıcı  
221307040  
yunusemiratici39@gmail.com

**Abstract—** This project develops a Library Automation System to manage book cataloging, member registration, and book lending/returns digitally. A modern web stack (React.js, Node.js/Express) is used for the user interface and API, while Prisma ORM and PostgreSQL handle the database. The database schema is fully normalized (up to 5NF) and includes tables for books, authors, categories, publishers, members, loans, and book-author relations. Key features include viewing available books, borrowing/returning books with automatic stock updates via database triggers, and tracking loan history. A SQL view is defined to list all currently active loans, and indexes on ISBN and member e-mail fields improve query performance.

## I. GİRİŞ

Günümüzde kütüphane işlemlerinin dijitalleştirilmesi, hem kütüphane personeli hem de kullanıcılar için işlemlerin hızlanması ve hata riskinin azaltılması açısından büyük önem taşımaktadır. Örneğin, manuel olarak yürütülen ödünç verme ve iade işlemleri zaman kaybına ve veri tutarsızlıklarına neden olabilir. Bu nedenle, üniversite kütüphanesinde kitap kayıtları, kullanıcı üyeliği, ödünç verme/iade ve stok takibi gibi işlemleri otomatize eden bir sistem geliştirilmesi amaçlanmıştır. Tasarlanan sistem; kitapkatalog yönetimi, kullanıcı etkileşimleri ve veritabanı katmanı olmak üzere üç temel bileşenden oluşur. Arayüzde React.js (Ant Design) kullanılmakta, sunucu tarafında Node.js ile Express.js web çerçevesi yer almakta ve veritabanı için PostgreSQL (Prisma ORM) tercih edilmektedir. Bu sayede sistem, kullanıcı dostu bir arayüzle, güncel web teknolojilerini kullanarak güvenilir bir şekilde çalışacak biçimde tasarlanmıştır.

## II. VERİ TABANI TASARIMI

Veritabanı Kitap, Yazar, Kategori, YayınEvi, Üye, Ödünç ve Kitap\_Yazar adlı ana tablolardan oluşmaktadır. Bu tablolar arasındaki ilişkiler aşağıdaki gibidir: - Kitap (ISBN PK, başlık, kategori\_id FK, yayın\_evi\_id FK, stok\_adi, vb.): Her kitap bir kategoriye ve bir yayın evine aittir (çoktan bire ilişki). - Yazar (yazar\_id PK, ad, soyad, vb.): Yazar bilgilerini tutar. - Kitap\_Yazar (kitap\_isbn FK, yazar\_id FK): Kitap-Yazar arasında çoktan çoğa ilişkiyi sağlar (bir kitap birden çok yazara; bir yazar birden çok kitaba sahip olabilir). ISBN ve yazar\_id birlikte birincil anahtar veya birleşik bir benzersiz anahtar oluşturur. - Kategori (kategori\_id PK, isim): Kitap kategorileri (örn. roman, biyografi) bilgilerini içerir. - YayınEvi (yayin\_evi\_id PK, ad, adres, vb.): Kitabın yayımlandığı yayınevi bilgilerini içerir. - Üye (uye\_id PK, isim, soyisim, eposta, şifre, vb.): Kütüphane üyelerinin bilgilerini depolar. E-posta alanı üzerinde benzersiz indeks tanımlıdır. - Ödünç (odunc\_id PK, uye\_id FK, kitap\_isbn FK, odunc\_tarihi, iade\_tarihi, teslim\_durumu, vb.): Kitap ödünç işlemleri için kayıtları içerir. Her ödünç kaydı bir üye ve bir

kitaba aittir. Tablolar arasındaki bu tasarım, ilişkisel veritabanı normalizasyon kurallarına uygun olarak veri tekrarını en aza indirecek şekilde düzenlenmiştir. Özellikle beşinci normal forma (5NF) kadar normalizasyon sağlanmıştır; yani her tablo yalnızca aday anahtarlarla ifade edilen birleştirme bağımlılıklarına izin verir. Bu sayede veri tutarlılığı ve esnekliği artırılmıştır. Örneğin, yazar bilgileri ayrı bir tabloda tutulduğu için aynı yazar bilgisi birden fazla kitap kaydında tekrarlanmaz. Veritabanında trigger kullanılarak stok kontrolü otomatikleştirilmiştir. Kitap ödünç verilince veya iade kaydı oluşturulunca tetikleyici aracılığıyla ilgili Kitap tablosundaki stok\_adi güncellenir. SQL tetikleyicileri, belirli bir kayıt ekleme/güncelleme/silme olayı gerçekleştiğinde otomatik olarak çalışan depolanmış işlemlerdir ve veri bütünlüğünü sağlamak için idealdir. Bu projede, örneğin yeni bir Ödünç kaydı eklendiğinde AFTER INSERT tetikleyicisi stoktan 1 düşürür; iade işlemi sırasında ise AFTER UPDATE veya DELETE tetikleyicisi stok ekleme yapar. Ayrıca, view olarak “aktif ödünçler” tablosu tanımlanmıştır; bu sanal tablo aracılığıyla şu anda teslim edilmemiş (aktif) ödünç kayıtları listelenebilir. Veritabanı olarak PostgreSQL kullanıldığından tetikleyiciler ve güncellenebilir görünüm gibi gelişmiş özellikler desteklenir. Performansı artırmak amacıyla indeks stratejileri uygulanmıştır. Özellikle ISBN alanı ve üye eposta alanı üzerinde indeks tanımlanmıştır. Bu alanların her biri yüksek seçiciliğe sahiptir (ISBN kitaplar için eşsizdir, üyelerin eposta adresleri benzersizdir) ve aramalarda anahtar sorguları hızlandırır. Örneğin bir kitap sorgulandığında ISBN üzerinden hızlı erişim sağlanır, bir üyenin bilgisi eposta ile sorgulandığında indeks sayesinde yanıt süresi azalır. İndeks kullanımı, sorgu performansı için kritik olan bu tip sütunlarda büyük kazanç sağlar.

## III. KULLANILAN TEKNOLOJİLER

- **React.js** – Kullanıcı arayüzü için seçilmiştir. React, bileşen tabanlı bir JavaScript kütüphanesidir ve “kullanıcı arayüzleri oluşturmak için serbest ve açık kaynaklı bir ön yüz JS kütüphanesi” olarak tanımlanır.
- **Ant Design** – React için bir UI bileşen kütüphanesidir. Ant Design şablonuna uygun yüksek kaliteli bileşenler içerir ve kurumsal uygulamalar için tasarlanmıştır. Hızlı ve tutarlı bir tasarım için tablo, form, buton gibi hazır elementler sunar.
- **Node.js** – Sunucu tarafı JavaScript çalışma zamanıdır. Node.js, “açık kaynaklı, çok platformlu bir JavaScript çalışma zamanı ortamı” olup, asenkron ve olay odaklı (event-driven) yapısıyla yüksek ölçeklenebilir web uygulamaları geliştirilmesine uygundur.

- **Express.js** – Node.js üzerinde çalışan hafif ve esnek bir web çatısıdır. Express, RESTful API geliştirmeyi kolaylaştırır ve minimal yapıdadır . Bu projede HTTP isteklerini yönlendirme ve işleme için kullanılmıştır.
- **Prisma (ORM)** – Veritabanı işlemleri için bir ORM aracıdır. “Prisma, Node.js ve TypeScript için modern, açık kaynak kodlu bir ORM aracıdır”. Prisma, şema tabanlı veri modelleme, otomatik veri türü güvenliği ve kolay migration desteği sağlar. Node.js/Express katmanında kitap, üye ve ödünç sorguları Prisma üzerinden tanımlıdır.
- **PostgreSQL** – İlişkisel veritabanı yönetim sistemidir. Güçlü SQL standart desteği ve zengin özellik setiyle bilinir . Projede kitap, üye, ödünç gibi veriler PostgreSQL’de depolanır.

#### IV. ÖZELLİKLER

Sistemde kullanıcılar öncelikle üye kaydı oluşturur ve kimlik doğrulama ile giriş yapar. React tabanlı arayüzde kütüphanedeki kitaplar listelenir; kullanıcılar kategori veya başlığa göre arama yapabilir. Her kitap için başlık, yazar(lar), kategori ve stok durumu gösterilir. Kullanıcı, elinde mevcut olmayan bir kitabı ödünç alabilir. Ödünç alma işleminde arayüzden istenen kitap seçilir ve onaylandığında sistem arka planda bir POST /odunc API çağrısı ile yeni bir ödünç kaydı oluşturur. Bu kayıt eklendiğinde PostgreSQL tetikleyicisi stok\_adedini otomatik olarak günceller. İade işlemi benzer şekilde gerçekleştirilir; kullanıcı ilgili ödünç kaydını işaretleyerek iade ettiğinde tetikleyici stok değerini geri artırır. Tüm ödünç işlemleri ödünç geçmişi olarak veritabanında saklanır; böylece geçmişte alınan iade edilen kitap kayıtları kalıcı olarak tutulur. Güncel aktif ödünçler için tanımlı olan SQL görünümü ile (örneğin `CREATE VIEW AktifOdunc AS SELECT \* FROM Odunc WHERE teslim\_durumu='EM'Durumla not completed') sadece iade edilmemiş kayıtlar listelenir. Verilen view, hem kullanıcıların mevcut ödünçleri görmesini sağlar hem de stok kontrolünün şeffaflığını artırır. Veritabanına erişim katmanında Prisma kullanılması, sorguların tip güvenliği içinde yazılmasını ve migration desteğini kolaylaştırır. Örneğin kitap sorgusu prisma.kitap.findMany({ where: { stok\_adedi: { gt: 0 } } }) şeklinde basitçe yapılır. Express.js ,React arayüzü ile JSON üzerinden haberleşir. Kullanıcı

arayüzünde Ant Design bileşenleri (Tablolar, Formlar, Butonlar) sayesinde modern ve uyumlu bir görünüm elde edilmiştir. Özetle, sistem kullanıcı kaydı, kitap listesi görüntüleme, ödünç alma/iade, stok güncelleme gibi işlevleri bütünlük olarak sunar. İndeksler sayesinde kitap arama ve üye doğrulama işlemleri hızlı çalışır; view ve trigger kullanımı ise veri bütünlüğü ve otomasyon açısından kritik rolleri üstlenir.

#### V. SONUÇ

Bu projede tasarlanan Kütüphane Otomasyon Sistemi, kütüphane işlemlerini tamamen dijital ortama taşıyarak operasyonel verimliliği artırmayı başarmıştır. Modern web teknolojileri (React/Ant Design, Node.js/Express, Prisma, PostgreSQL) kullanılarak hem kullanıcı arayüzü hem de veritabanı katmanı güncel ve ölçeklenebilir biçimde uygulanmıştır. Veritabanı tasarımı beşinci normal forma uygun olarak yapılmış, tekrarlı verilerden kaçınılmış ve veri bütünlüğü korunmuştur . Ayrıca tetikleyiciler ve görünümle otomatik stok kontrolü ve aktif ödünç listesi gibi özellikler eklenmiştir. Sonuç olarak geliştirilen sistem, kütüphane yönetimini kolaylaştırmakta ve ileride yeni modüller (ör. ceza yönetimi, rezervasyon) eklemeye açık bir altyapıya sahiptir.

#### KAYNAKÇA

GeeksforGeeks, “System Design for Library Management”, 17 Temmuz 2024

PostgreSQL Documentation, “What Is PostgreSQL?”

GeeksforGeeks, “Best Backend Technologies for React in 2025” (Node.js, Express.js, Prisma, React)

Database normalization  
- Wikipedia [https://en.wikipedia.org/wiki/Database\\_normalization](https://en.wikipedia.org/wiki/Database_normalization)

PostgreSQL: Documentation: 17: 1. What Is PostgreSQL?  
<https://www.postgresql.org/docs/current/intro-what.html>

Ant Design of React  
Ant Design <https://ant.design/docs/react/introduce/>