

**“Kitaplık”**

*BİL372 Veritabanı Sistemleri Projesi*

**Son Rapor**

**Ali Doğan Çamur 231101017**

**Ege Azca 231101057**

**Yankı Yağız Ozan 231101059**

## 1. Domain Gerçek Dünya Probleminin Tanımı

Projemiz, insanların sahip olduğu 2. el kitapları tekrar kullanışlı hale getirmeyi hedeflemektedir. Kampüsümüzde pek çok öğrencinin, okudukları veya artık kullanmadıkları ders kitaplarını yurt odalarında ya da evlerinde beklettiğini gözlemliyoruz. Bu durum aynı kitaplara ihtiyaç duyan diğer öğrencilerin bu kaynaklara erişmek için yüksek ücretler ödemek zorunda kalmasına neden olmaktadır. Aynı zamanda basılı kitaplara erişimi kısıtlı olan veya her dönem yeni kitap alacak bütçesi olmayan öğrencilerimiz için de ciddi zorluklar yaratmaktadır.

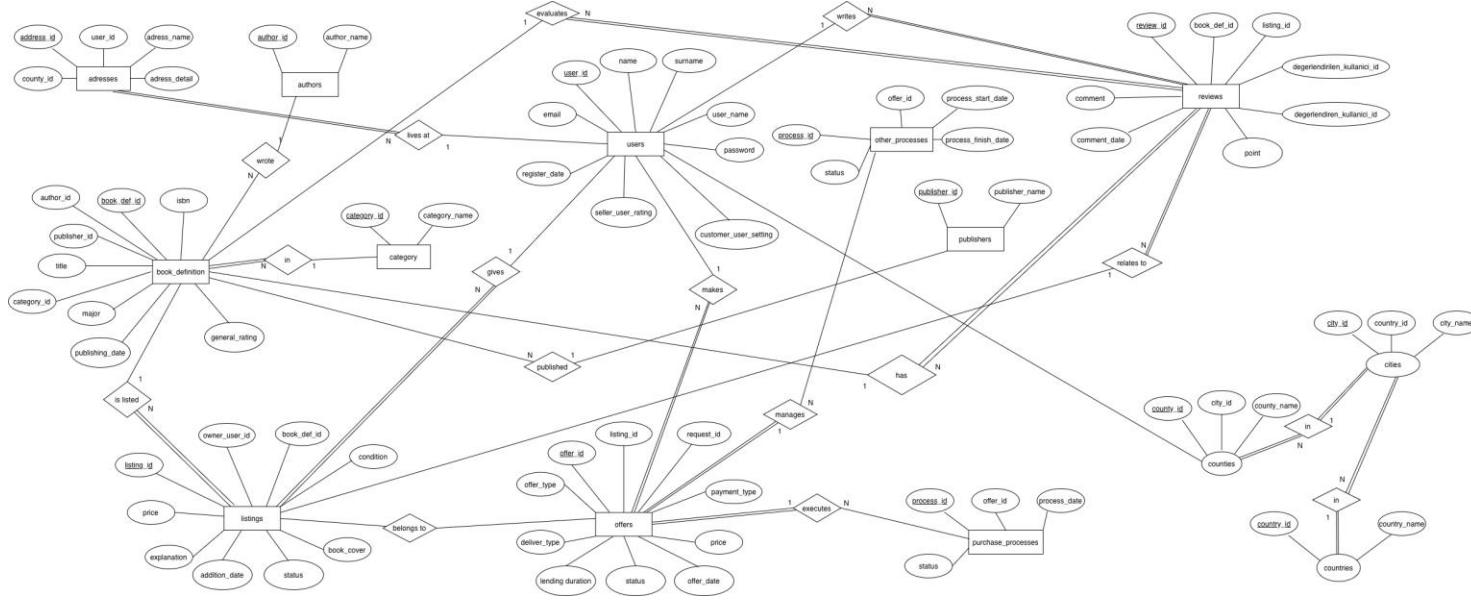
Platformumuz, 2. el kitap satın alım ve kiralama platformudur. Kullanıcılar bu platform üzerinden ihtiyaçları olan kitabı aratabilecek ve dış piyasaya kıyasla çok daha uyguna diğer kullanıcıların 2. el kitaplarını satın alabilecek veya bir süreliğine kiralayabileceklerdir. Sisteme yüklenen kitapların kullanılma derecelerine, ilk yüklenme fiyatına göre göre alıcı satıcıya teklifler sunabilecektir.

## 2. Gereksinim Analizi

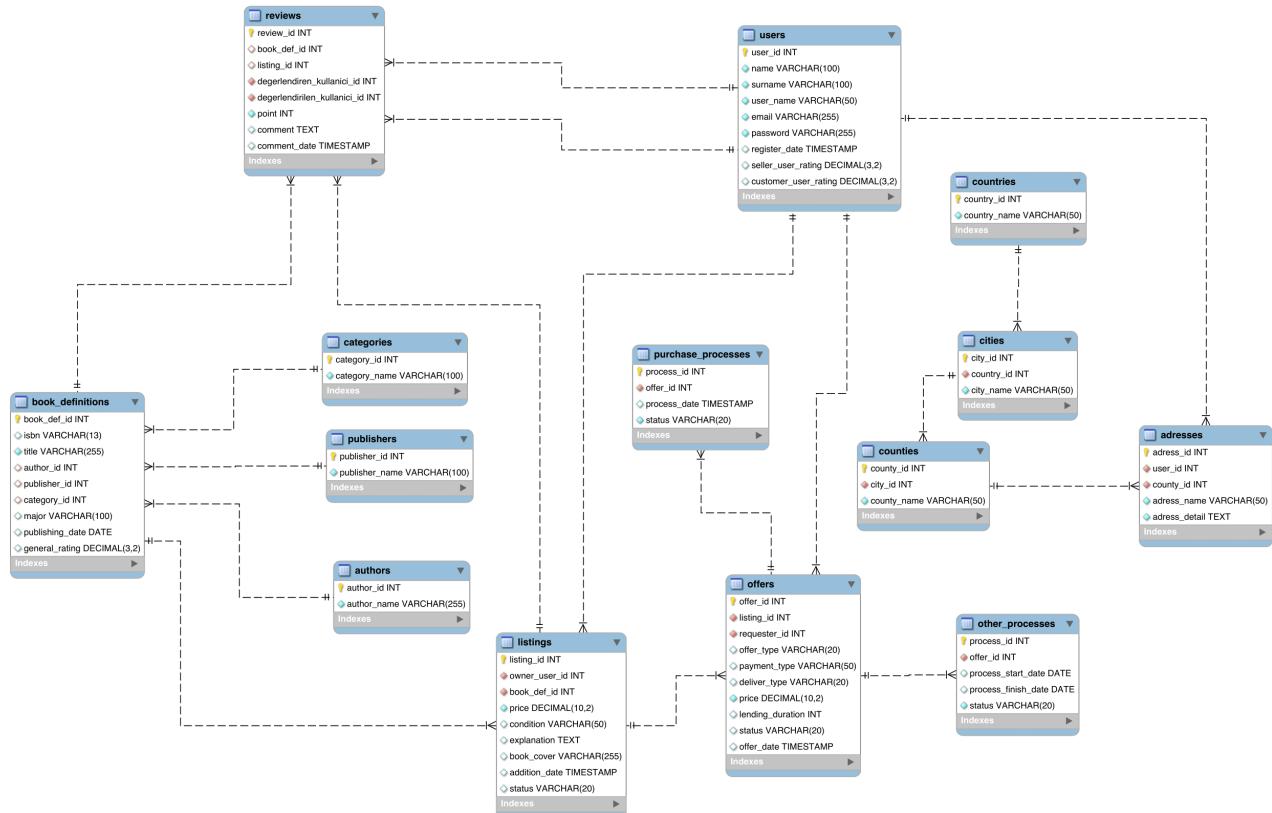
*Sistemin temel işlevsel gereksinimleri şunlardır:*

- Kullanıcı Yönetimi:** Öğrenciler sisteme kaydolabilmeli, giriş yapabilmeli ve profil bilgilerini (adres vb.) yönetebilmelidir.
- Katalog Yapısı:** Kitaplar bir kez tanımlanmalı (ISBN, Yazar, Yayınevi), kullanıcılar bu tanımlar üzerinden ilan açmalıdır.
- İlan Yönetimi:** Kullanıcılar ellerindeki kitaplar için kondisyon ve fiyat belirterek "Satılık", "Kiralık" veya "Ödünç" ilanı açabilmelidir.
- Teklif Sistemi:** Alıcılar, ilanlara fiyat teklifi verebilmeli, kiralama süresi talep edebilmelidir.
- Değerlendirme Sistemi:** İşlem sonunda taraflar birbirini ve kitabı puanlayabilmelidir. Bu puanlar "Satıcı Puanı" ve "Müşteri Puanı" olarak profile yansımalıdır.

### 3. Kavramsal Tasarım (EER Diyagramı)



#### 4. Mantıksal Tasarım ve Şema Diyagramları



## 5. Tasarımın Uygulanması

### 5.1. Kullanılan Teknolojiler ve Frameworkler

Proje 3 Katmanlı Mimari kullanılarak geliştirilmiştir.

1. **Veritabanı (VTYS): MySQL**
2. **Backend: Node.js + Express.js**
3. **Frontend: JavaScript + HTML/CSS**

### 5.2. Tabloların Oluşturulması

```
1.  create table countries (
2.    country_id int auto_increment primary key,
3.    country_name varchar(50) not null unique
4. );
5.  create table cities (
6.    city_id int auto_increment primary key,
7.    country_id int not null,
8.    city_name varchar(50) not null,
9.    foreign key (country_id) references countries(country_id) on delete cascade
10. );
11. create table counties (
12.    county_id int auto_increment primary key,
13.    city_id int not null,
14.    county_name varchar(50) not null,
15.    foreign key (city_id) references cities(city_id) on delete cascade
16. );
17. create table authors (
18.    author_id int auto_increment primary key,
19.    author_name varchar(255) not null unique
20. );
21.
22. create table publishers (
23.    publisher_id int auto_increment primary key,
24.    publisher_name varchar(100) not null unique
25. );
26.
27. create table categories (
28.    category_id int auto_increment primary key,
29.    category_name varchar(100) not null unique
30. );
31. -- 2. kullanıcılar tablosu
32. create table users (
33.    user_id int auto_increment,
34.    name varchar(100) not null,
35.    surname varchar(100) not null,
36.    user_name varchar(50) not null,
37.    email varchar(255) not null,
38.    `password` varchar(255) not null,
39.    register_date timestamp default current_timestamp,
40.    seller_user_rating decimal(3, 2) default 0.00,
41.    customer_user_rating decimal(3, 2) default 0.00,
42.    primary key (user_id),
43.    unique (user_name),
44.    unique (email)
45. );
46.
47. -- 3. kitap tanımları (katalog)
48. create table book_definitions (
49.    book_def_id int auto_increment,
50.    isbn varchar(13) unique,
51.    title varchar(255) not null,
52.    author_id int,
53.    publisher_id int,
54.    category_id int,
55.    major varchar(100),
56.    publishing_date date,
57.    general_rating decimal(3, 2) default 0.00,
58.    primary key (book_def_id),
59.    foreign key (author_id) references authors(author_id),
60.    foreign key (publisher_id) references publishers(publisher_id),
61.    foreign key (category_id) references categories(category_id)
62. );
63.
64.
```

```

65. -- 4. ilanlar
66. create table listings (
67.     listing_id int auto_increment,
68.     owner_user_id int not null,
69.     book_def_id int not null,
70.     `condition` varchar(50),
71.     explanation text,
72.     book_cover varchar(255),
73.     addition_date timestamp default current_timestamp,
74.     status varchar(20) default 'Yayinda',
75.     price decimal(10, 2) not null default 0.00,
76.     primary key (listing_id),
77.     foreign key (owner_user_id) references users(user_id) on delete cascade,
78.     foreign key (book_def_id) references book_definitions(book_def_id)
79. );
80. -- 5. teklifler
81. create table offers (
82.     offer_id int auto_increment,
83.     listing_id int not null,
84.     requester_id int not null,
85.     offer_type varchar(20),
86.     payment_type varchar(50),
87.     deliver_type varchar(20),
88.     price decimal(10, 2) not null default 0.00,
89.     lending_duration int,
90.     status varchar(20) default 'Aktif',
91.     offer_date timestamp default current_timestamp,
92.     primary key (offer_id),
93.     foreign key (listing_id) references listings(listing_id) on delete cascade,
94.     foreign key (requester_id) references users(user_id) on delete cascade
95. );
96.
97. -- 6. islem tablolari
98. create table purchase_processes (
99.     process_id int auto_increment,
100.    offer_id int not null,
101.    process_date timestamp default current_timestamp on update current_timestamp,
102.    status varchar(20) not null default 'Talep Edildi',
103.    primary key (process_id),
104.    unique (offer_id),
105.    foreign key (offer_id) references offers(offer_id) on delete cascade
106. );
107.
108. create table other_processes (
109.     process_id int auto_increment,
110.    offer_id int not null,
111.    process_start_date date,
112.    process_finish_date date,
113.    status varchar(20) not null default 'Talep Edildi',
114.    primary key (process_id),
115.    unique (offer_id),
116.    foreign key (offer_id) references offers(offer_id) on delete cascade
117. );
118.
119. -- 7. adresler
120. create table addresses (
121.     address_id int auto_increment,
122.     user_id int not null,
123.     county_id int not null,
124.     address_name varchar(50) not null,
125.     address_detail text not null,
126.     primary key (address_id),
127.     foreign key (user_id) references users(user_id) on delete cascade,
128.     foreign key (county_id) references counties(county_id)
129. );
130.
131. -- 8. degerlendirmeler
132. create table reviews (
133.     review_id int auto_increment,
134.     book_def_id int,
135.     listing_id int,
136.     degerlendiren_kullanici_id int not null,
137.     degerlendirilen_kullanici_id int not null,
138.     point int not null check (point >= 1 and point <= 5),
139.     `comment` text,
140.     comment_date timestamp default current_timestamp,
141.     primary key (review_id),
142.     foreign key (book_def_id) references book_definitions(book_def_id),
143.     foreign key (listing_id) references listings(listing_id),
144.     foreign key (degerlendiren_kullanici_id) references users(user_id),
145.     foreign key (degerlendirilen_kullanici_id) references users(user_id)
146. );

```

### 5.3 Viewlar

```
1. create view view_catalog_summary as
2. select bd.title, a.author_name, bd.isbn, bd.general_rating, count(b.listing_id) as total_listings_available
3. from book_definitions bd
4. left join listings b on bd.book_def_id = b.book_def_id and b.status = 'Yayında'
5. left join authors a on bd.author_id = a.author_id
6. group by bd.book_def_id;
```

- ~ **Mantık:** Kitap tanımlarını ve o kitaba ait satışta kaç adet kopya olduğunu listeler.
- ~ **Neden Left Join?** Eğer WHERE kullanısaydık, hiç ilanı olmayan kitaplar sonuçtan elenirdi (çünkü status NULL olurdu). Koşulu JOIN içine alarak, "Sadece yayında olan ilanları bağla, eğer yoksa yine de kitabı liste ama ilan kısmını NULL yap" demiş oluyoruz

```
1. create view view_active_listings as
2. select b.listing_id, bd.title, u.user_name as seller, b.condition, b.status
3. from listings b
4. join book_definitions bd on b.book_def_id = bd.book_def_id
5. join users u on b.owner_user_id = u.user_id
6. where b.status = 'Yayında';
```

- ~ **Mantık:** Sadece şu an satın alınabilir durumdaki ilanları düz bir liste olarak sunar.

```
1. create view view_full_addresses as
2. select a.adress_id, u.user_name, a.adress_name, a.adress_detail, cnt.county_name, ci.city_name,
   co.country_name
3. from addresses a
4. join users u on a.user_id = u.user_id
5. join counties cnt on a.county_id = cnt.county_id
6. join cities ci on cnt.city_id = ci.city_id
7. join countries co on ci.country_id = co.country_id;
```

- ~ **Mantık:** Normalizasyon gereği parçalanmış coğrafi veriyi (county\_id -> city\_id-> country\_id) tek bir okunabilir satır'a çevirir.
- ~ **Tasarım Tercihi (Zincirleme Inner Join):** Adres hiyerarşisinin tam olduğunu garanti eder. Üstelik tek tablo gibi de gözükebilir.

```
1. create view view_active_offers as
2. select o.offer_id, bd.title, u.user_name as requester, o.offer_type, o.price, o.status
3. from offers o
4. join listings b on o.listing_id = b.listing_id
5. join book_definitions bd on b.book_def_id = bd.book_def_id
6. join users u on o.requester_id = u.user_id
7. where o.status = 'Aktif';
```

- ~ **Mantık:** Sadece üzerinde aksiyon alınması gereken (Bekleyen) teklifleri gösterir. Reddedilen veya onaylananlar elenir.

### 5.4 Veri Yükleme

Mock veriler Mockaroo ve kendimiz yazdığımız queryler ile yüklenmiştir.

### 5.5 İndisler

```
1. create index idx_book_title on book_definitions(title);
```

- ~ **Amaç:** Kitap ismine göre yapılan aramaları ve sıralamaları hızlandırmak.

```
1. create index idx_book_isbn on book_definitions(isbn);
```

- ~ **Amaç:** ISBN numarası üzerinden yapılan kesin eşleşme sorgularını optimize etmek.

```
1. create index idx_user_name on users(name, surname);
```

- ~ **Amaç:** Kullanıcıları isim ve soyisim kombinasyonuyla aramak.

```
1. create index idx_offer_status on offers(status);
```

- ~ **Amaç:** Belirli bir durumdaki (Örn: 'Aktif') teklifleri filtrelemek.

## 5.6 Soru Tasarımları

### 1. GET /api/catalog:

```
SELECT bd.book_def_id, bd.isbn, bd.title, a.author_name, bd.general_rating,
       COUNT(CASE WHEN b.status = 'Yayında' THEN 1 END) AS total_listings_available
  FROM book_definitions bd
  LEFT JOIN listings b ON bd.book_def_id = b.book_def_id
  LEFT JOIN authors a ON bd.author_id = a.author_id
 WHERE ... GROUP BY bd.book_def_id
```

- ~ **Amaç:** Kütüphane genelindeki kitap tanımlarını listelemek.
- ~ **Neden Left Join:** Hiç satıcısı olmayan (stokta olmayan) kitapların da arama sonucunda çıkışmasını sağlar. INNER JOIN olsaydı sadece satılan kitaplar listelenirdi.

### 2. GET /api/listings:

```
SELECT l.listing_id, bd.title, u.user_name AS seller, u.user_id AS seller_user_id, l.condition, l.status,
       l.price, bd.general_rating
  FROM listings l
  JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
  JOIN users u ON l.owner_user_id = u.user_id
  LEFT JOIN authors a ON bd.author_id = a.author_id
 WHERE l.status = 'Yayında'
```

- ~ **Amaç:** Satıştaki fiziksel ürünlerini listelemek.
- ~ **Neden Inner Join:** Kitap tanımı (bd) veya satıcısı (u) olmayan bir ilan "bozuk veri"dir. Bunların listelenmesini engellemek için INNER JOIN ile veri bütünlüğü zorlandı.

### 3. GET /api/listings/:id:

```
SELECT l.*, bd.title, u.user_name, u.seller_user_rating AS seller_user_rating, u.user_id AS owner_user_id
  FROM listings l
  JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
  JOIN users u ON l.owner_user_id = u.user_id
 WHERE l.listing_id = ?
```

- ~ **Amaç:** Tek bir ilanın detay sayfasını göstermek.

### 4. POST /api/offers:

```
SELECT l.*, bd.title, u.user_name, u.seller_user_rating AS seller_user_rating, u.user_id AS owner_user_id
```

- ~ **Amaç:** İlanın sahibini bulmak.

### 5. POST /api/offers/:id/accept:

```
SELECT l.*, bd.title, u.user_name, u.seller_user_rating AS seller_user_rating, u.user_id AS owner_user_id
```

- ~ **Amaç:** Transaction başlatıldığında, onaylanan teklifin hangi ilana (listing\_id) ait olduğunu bulmak.

### 6. GET /api/offers/:id:

```
SELECT o.*, bd.title, u.user_name AS requester_name, u.user_id AS requester_id
  FROM offers o
  JOIN listings l ON o.listing_id = l.listing_id
  JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
  JOIN users u ON o.requester_id = u.user_id
 WHERE o.offer_id = ?
```

- ~ **Amaç:** Bir teklifin özetini göstermek.
- ~ **Tasarım:** Inner Join zinciri. Teklif -> İlan -> Kitap -> Teklifi Veren Kullanıcı. Bu zincirden biri bile eksikse teklif verisi bozuktur, getirilmez.

## 7. GET /api/active-offers:

```
SELECT * FROM view_active_offers
```

- ~ **Amaç:** Aktif teklifleri listelemek.

## 8. POST /api/login:

```
SELECT * FROM users WHERE email = ? OR user_name = ?
```

- ~ **Amaç:** Kullanıcıyı bulmak.
- ~ **Tasarım:** OR kullanımı. Kullanıcının hem e-posta hem de kullanıcı adıyla giriş yapabilmesini sağlar.

## 9. POST /api/register:

```
SELECT * FROM users WHERE email = ? OR user_name = ? LIMIT 1
```

- ~ **Amaç:** Email veya kullanıcı adı dolu mu?
- ~ **Tasarım:** *LIMIT 1*. Performans için. İlk eşleşmeyi bulunca aramayı durdurur.

## 10. GET /api/user/username/:username:

```
SELECT * FROM users WHERE user_name = ?
```

- ~ **Amaç:** Kullanıcı adına göre profil bulmak.

## 11. GET /api/user/:id:

```
SELECT * FROM users WHERE user_id = ?
```

- ~ **Amaç:** Kullanıcı idlerine göre profil bulmak.

## 12. GET /api/user/:id:

```
SELECT l.*, bd.title FROM listings l JOIN book_definitions bd ON l.book_def_id = bd.book_def_id WHERE l.owner_user_id = ?
```

- ~ **Amaç:** Kullanıcının sattığı ürünleri listelemek.

## 13. GET /api/user/:id:

```
SELECT o.*, bd.title, u.user_name as requester_name, u.user_id as requester_id  
FROM offers o  
JOIN listings l ON o.listing_id = l.listing_id  
JOIN book_definitions bd ON l.book_def_id = bd.book_def_id  
JOIN users u ON o.requester_id = u.user_id  
WHERE o.offer_id = ?
```

- ~ **Amaç:** Bana gelen teklifleri görmek.
- ~ **Tasarım:** Teklif bana değil, benim ilanımı gelir. Bu yüzden Teklif -> İlan -> İlan Sahibi (Ben) zinciri kurulmuştur.

## 14. GET /api/user/:id/public

```
SELECT user_id, user_name, name, surname, seller_user_rating, customer_user_rating FROM users WHERE user_id = ?
```

- ~ **Amaç:** Başkası profili gezerken gösterilecek bilgi.
- ~ **Tasarım:** Özel kolon seçimi (SELECT user\_id). Şifre ve email gibi hassas verileri gizlemek için SELECT \* kullanılmadı.

## 15. GET /api/user/:id/public

```
SELECT o.*, bd.title, u.user_name as requester_name, u.user_id as requester_id
FROM offers o
JOIN listings l ON o.listing_id = l.listing_id
JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
JOIN users u ON o.requester_id = u.user_id
WHERE o.offer_id = ?
```

- ~ Amaç: O kişinin şu an sattığı ürünleri göstermek.

## 16. GET /api/user/:id/public

```
SELECT r.*, u.user_name as reviewer_name, u.user_id as reviewer_id, bd.title as book_title, l.listing_id
FROM reviews r
JOIN users u ON r.değerlendiren_kullanici_id = u.user_id
LEFT JOIN book_definitions bd ON r.book_def_id = bd.book_def_id
LEFT JOIN listings l ON r.listing_id = l.listing_id
WHERE r.değerlendirilen_kullanici_id = ?
ORDER BY r.comment_date DESC
```

- ~ Amaç: Kişinin aldığı yorumları göstermek.
- ~ Neden Left Join: Eğer yorum yapılan kitap veya ilan silinmişse bile, yorumun (puanın) profilde kalması için.

## 17. PUT /api/user/:id

```
SELECT * FROM users WHERE (email = ? OR user_name = ?) AND user_id != ? LIMIT 1DESC
~ Amaç: Yeni girilen email başkasının mı?
```

## 18. GET /api/user/:id/orders

```
SELECT o.offer_id, l.listing_id, bd.book_def_id,
       l.owner_user_id AS seller_user_id,
       bd.title AS book_title,
       o.status, o.offer_date AS process_date, o.price
FROM offers o
JOIN listings l ON o.listing_id = l.listing_id
JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
WHERE o.requester_id = ?
ORDER BY o.offer_date DESC
```

- ~ Amaç: Bu sorgu, giriş yapmış kullanıcının Alıcı (Requester) rolünde olduğu işlemleri listeler. Kullanıcının profilindeki "Siparişlerim" veya "Yaptığım Teklifler" ekranını dolduran ana sorgudur.
- ~ Tasarım: Sistemde "Var olmayan bir ilanın teklifi de geçersizdir" olarak kabul edilmiştir. Eğer bir ilan sistemden fiziksel olarak silinmişse ama kullanıcı o ilana geçmişte teklif vermişse dahı bu listede göremez.

## 19. POST /api/books

```
SELECT * FROM ${table} WHERE ${col} = ? LIMIT 1
~ Amaç: Yazar/Yayinevi/Kategori veritabanında var mı diye bakar (Lookup). Varsa ID'sini döner, yoksa yaratır.
```

## 20. GET /api/user/:id/addresses

```
SELECT r.*, u.user_name as reviewer_name, u.user_id as reviewer_id, bd.title as book_title, l.listing_id
FROM reviews r
JOIN users u ON r.değerlendiren_kullanici_id = u.user_id
LEFT JOIN book_definitions bd ON r.book_def_id = bd.book_def_id
LEFT JOIN listings l ON r.listing_id = l.listing_id
WHERE r.değerlendirilen_kullanici_id = ?
ORDER BY r.comment_date DESC
```

- ~ Amaç: Adres listesini okunabilir formatta getirmek.

## 21. GET /api/addresses/:id

```
SELECT a.*, cnt.county_name, ci.city_name, co.country_name FROM addresses a JOIN counties cnt ... JOIN cities
ci ... JOIN countries co ... WHERE a.adress_id = ?
~ Amaç: Düzenleme ekranı için tek bir adresi getirmek.
```

## 22. POST /api/reviews

```
SELECT r.*, u.user_name as reviewer_name, u.user_id as reviewer_id, bd.title as book_title, l.listing_id
FROM reviews r
JOIN users u ON r.değerlendiren_kullanici_id = u.user_id
LEFT JOIN book_definitions bd ON r.book_def_id = bd.book_def_id
LEFT JOIN listings l ON r.listing_id = l.listing_id
WHERE r.değerlendirilen_kullanici_id = ?
ORDER BY r.comment_date DESC
```

- ~ Amaç: Bu ilana zaten yorum yapmış mı'nın kontrolü.

## 23. POST /api/reviews

```
SELECT * FROM reviews WHERE book_def_id = ? AND değerlendiren_kullanici_id = ? AND listing_id IS NULL LIMIT 1
```

- ~ Amaç: Bu kitaba genel bir yorum yapmış mı?

## 24. POST /api/reviews

```
SELECT AVG(point) AS avggp FROM reviews WHERE book_def_id = ?
```

- ~ Amaç: Kitabın yeni ortalama puanını hesaplamak.

## 25. POST /api/reviews

```
SELECT AVG(point) AS avg_seller FROM reviews WHERE değerlendirilen_kullanici_id = ? AND listing_id IS NOT NULL
```

- ~ Amaç: Kullanıcının "Satıcı" kimliğiyle aldığı puanların ortalamasını hesaplamak.

## 26. POST /api/review

```
SELECT AVG(point) AS avg_customer FROM reviews WHERE değerlendirilen_kullanici_id = ? AND listing_id IS NULL
```

- ~ Amaç: Kullanıcının "Normal Kullanıcı" (kitap yorumcusu) olarak aldığı puanları hesaplamak.

## 27. GET /api/reviews

```
SELECT * FROM reviews WHERE listing_id = ?
```

- ~ Amaç: Belirli bir satış işleminin yorumlarını getirmek.

## 28. GET /api/reviews

```
SELECT * FROM reviews WHERE book_def_id = ?
```

- ~ Amaç: Belirli bir kitabın yorumlarını getirmek.

## 29. GET /api/reviews

```
SELECT * FROM reviews ORDER BY comment_date DESC LIMIT 100
```

- ~ Amaç: Anasayfa veya akışta son yorumları göstermek.

## 30. GET /api/user/:id/reviews

```
SELECT r.*, u.user_name as reviewer_name, u.user_id as reviewer_id, bd.title as book_title, l.listing_id
FROM reviews r
JOIN users u ON r.değerlendiren_kullanici_id = u.user_id
LEFT JOIN book_definitions bd ON r.book_def_id = bd.book_def_id
LEFT JOIN listings l ON r.listing_id = l.listing_id
WHERE r.değerlendirilen_kullanici_id = ?
ORDER BY r.comment_date DESC
```

- ~ Amaç: Profil sayfasında "Yorumlarım" sekmesi.

- ~ Neden Left Join: Yorumun ait olduğu nesne silinse bile yorumu göstermek için.

## 31. GET /api/countries

```
SELECT * FROM countries
```

- ~ Amaç: Ülke dropdown listesi.

### 32. GET /api/cities/:countryId

```
SELECT * FROM cities WHERE country_id = ?
```

~ Amaç: Seçilen ülkeye göre şehirleri getirmek.

### 33. GET /api/counties/:cityId

```
SELECT * FROM cities WHERE city_id = ?
```

~ Amaç: Seçilen ülkeye göre ilçeleri getirmek.

### 34. GET /api/authors

```
SELECT * FROM authors ORDER BY author_name
```

~ Amaç: Yazar dropdown listesi (Alfabetik).

### 35. GET /api/publishers

```
SELECT * FROM publishers ORDER BY publisher_name
```

~ Amaç: Yayınevi dropdown listesi (Alfabetik).

### 36. GET /api/categories

```
SELECT * FROM categories ORDER BY category_name
```

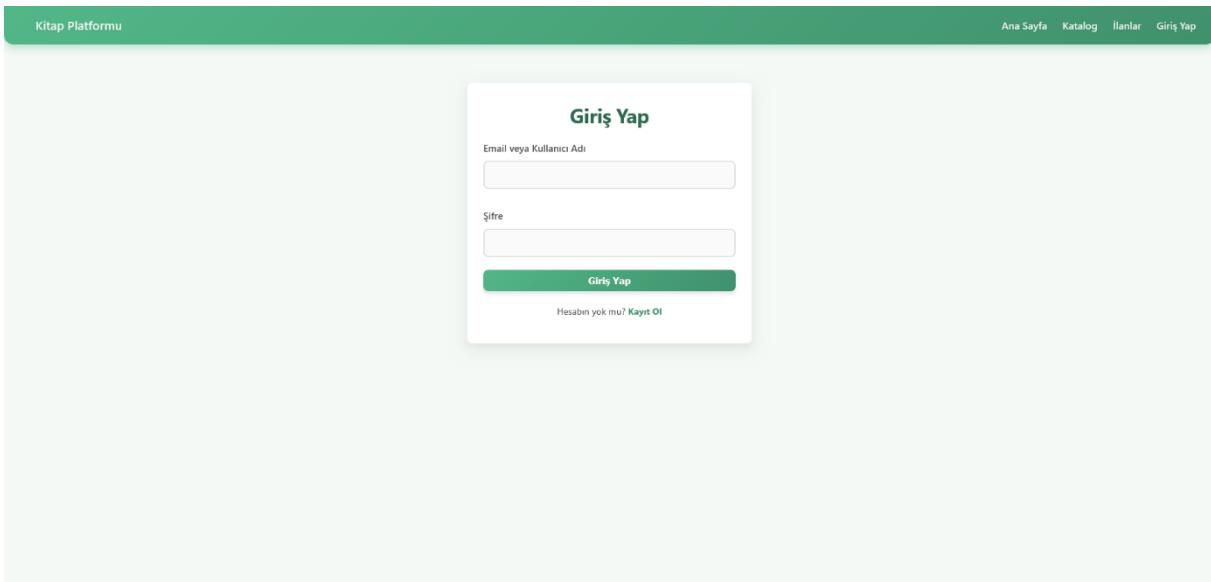
~ Amaç: Kategori dropdown listesi (Alfabetik).

### 37. GET /api/user/:id/sales

```
SELECT o.offer_id, l.listing_id, bd.book_def_id, bd.title, o.status,
       o.offer_date as process_date, o.price,
       u.user_name as buyer_name, u.user_id as buyer_user_id
FROM offers o
JOIN listings l ON o.listing_id = l.listing_id
JOIN book_definitions bd ON l.book_def_id = bd.book_def_id
JOIN users u ON o.requester_id = u.user_id
WHERE l.owner_user_id = ?
      AND o.status = 'Onaylandı'
      AND o.offer_type LIKE '%sat%'
ORDER BY o.offer_date DESC
```

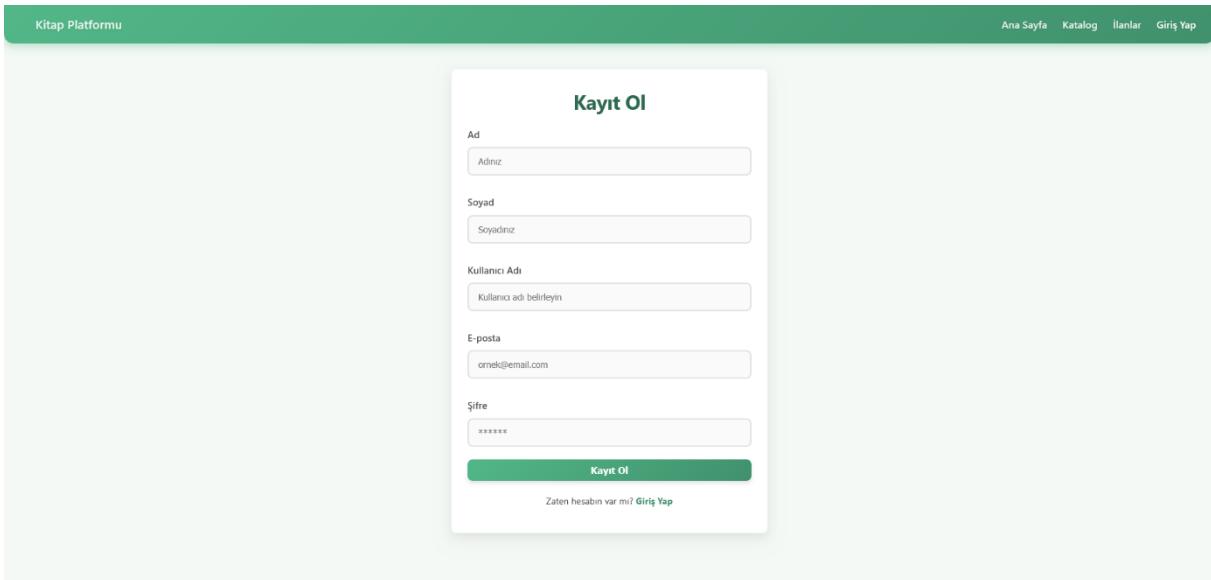
~ Amaç: Bu sorgu, giriş yapmış kullanıcının Satıcı olduğu ve satış işleminin başarıyla tamamlandıği geçmiş kayıtları listeler.

## 7. Uygulama Program Tanıtımı:



The screenshot shows the login page of a web application. At the top, there is a green header bar with the text "Kitap Platformu" on the left and "Ana Sayfa" (Home), "Katalog" (Catalog), "İlanlar" (Ads), and "Giriş Yap" (Log In) on the right. Below the header is a white login form titled "Giriş Yap". It contains two input fields: "Email veya Kullanıcı Adı" (Email or User Name) and "Şifre" (Password). Below these fields is a green "Giriş Yap" button. At the bottom of the form, there is a link "Hesabın yok mu? Kayıt Ol" (No account? Sign Up).

Giriş Ekranı



The screenshot shows the sign-up page of the application. At the top, there is a green header bar with the text "Kitap Platformu" on the left and "Ana Sayfa" (Home), "Katalog" (Catalog), "İlanlar" (Ads), and "Giriş Yap" (Log In) on the right. Below the header is a white sign-up form titled "Kayıt Ol". It contains six input fields: "Ad" (Name), "Soyad" (Last Name), "Kullanıcı Adı" (User Name), "E-posta" (Email), "Şifre" (Password), and a password confirmation field. Below these fields is a green "Kayıt Ol" button. At the bottom of the form, there is a link "Zaten hesabın var mı? Giriş Yap" (Already have an account? Log In).

Kayıt Olma Ekranı

## Kitap Al / Sat / Kirala

[Q. Kataloğa Git](#)[İlanlara Bak](#)[+ İlan Ekle](#)

## Editörün Seçimleri



## Ince Memed

Yaşar Kemal



## Zamanın Kısa Tarihi

Stephen Hawking

[Tümünü Gör →](#)

## Ana Ekran

## Tüm İlanlar

## Filtreler

Arama

Kitap veya yazar...

Sırala

Yeniye Göre (Yeni önce)

[Filtrele](#)

## Bilinmeyen Bir Kadının Mektubu

Satıcı: erenus

Puan: 0

Durum: Yeni

Fiyat: 280.00₺

[Teklif Ver](#)

## Bir Kurt Sevdim

Satıcı: erenus

Puan: 1

Durum: Yıpranmış

Fiyat: 30.00₺

[Teklif Ver](#)

## Kırk Mantolu Madonna

Satıcı: mehmetk

Puan: 4

Durum: Yeni

Fiyat: 140.00₺

[Teklif Ver](#)

## Harry Potter ve Felsefe Taşı

Satıcı: aysed

Puan: 4.7

Durum: Çok iyi

Fiyat: 250.00₺

## İlanlar

## Profil

### Kullanıcı Bilgileri

Kullanıcı Adı: ahmety2

Ad Soyad: Ahmet Yılmaz

Satıcı Puanı: ★ 4.00

Müşteri Puanı: ★ 0.00

[Satıcışlerim ve İşlemlerim](#)

[Bildirimler](#)

[Profilini Düzenle](#)

### Adreslerim

Ev

Kadıköy / İstanbul / Türkiye  
Moda Caddesi No:12

[Düzenle](#)

23123

Kadıköy / İstanbul / Türkiye  
q231

[Düzenle](#)

[Yeni Adres Ekle +](#)

### Aktif İlanlarım



Kürk Mantolu Madonna

Durum: İşlemde

[Yönet](#)



1984

Durum: Yayında

[Yönet](#)

[Yeni İlan Ekle +](#)

### Bana Gelen Teklifler

Bekleyen teklif yok.

### Bana Gelen Yorumlar

zeynepc

02.12.2025

Kitap: Kürk Mantolu Madonna

★ 4

1

## Profil Ekranı

Kitap Platformu

Ana Sayfa Katalog İlanlar Profil (ahmety2) Çıkış

### Teklif Ver

<b>Teklif Verdiğiniz İlan</b>	<b>Teklif Türü</b>
Kitap: Harry Potter ve Felsefe Taşı	Satın Alma
Satıcı: aysed	Teklif Ettığınız Fiyat (₺)
Durum: Çok İyi	Örm: 40
Liste Fiyatı: 250.00 ₺	Ödeme Yöntemi
	Nakit (Eden)
	Teslimat Yöntemi
	Eden Teslim

**Teklifi Gönder**

### Teklif Verme

Kitap Platformu

Ana Sayfa Katalog İlanlar Profil (ahmety2) Çıkış

### Adresleri Yönet

Adres Adı	Evin
Ülke	Türkiye
İl	Ankara
İlçe	Çankaya
Adres Detay	Söğütözü Mahallesi

**Kaydet**

### Adres Ekleme/Düzenleme Ekranı

## 7. Sonuç

Bu proje kapsamında, öğrencilerin kitap paylaşım sorununa çözüm getiren, ölçeklenebilir bir web uygulaması geliştirilmiştir. Veritabanı tasarımindan 3NF kurallarına sadık kalılmış, performans için indeksleme ve view yapıları kullanılmıştır. Backend tarafında RESTful mimari benimsenmiş, ve gerekli SQL sorgularıyla sistemin istenen isteklere tutarlı ve istediği şekilde yanıt vermesi sağlanmıştır.