

AKILLI KÜTÜPHANE YÖNETİM SİSTEMİ

PROJE RAPORU

Üniversite: Karadeniz Teknik Üniversitesi

Fakülte: Of Teknoloji Fakültesi

Bölüm: Yazılım Mühendisliği

Ders Adı: Veritabanı ve Yönetimi

Proje Adı: Akıllı Kütüphane Sistemi

Öğrenci Adı: Nehir Buket Polat

Öğrenci No: 445898

1-PROJENİN AMACI

Bu projenin amacı, bir kütüphanede gerçekleşen kitap yönetimi, kullanıcı yönetimi, ödünç alma-iade işlemleri ve ceza hesaplamalarının bilgisayar ortamında güvenli, düzenli ve merkezi bir sistem üzerinden yürütülmesini sağlamaktır.

Geliştirilen sistem:

- **Admin ve User (Kullanıcı)** rollerini ayırır
- Yetkilendirmeyi JWT (JSON Web Token) ile sağlar
- **RESTful API** mimarisini kullanır
- Frontend (HTML + JS) ile Backend (Flask) katmanlarını ayırır
- **PostgreSQL** veritabanı üzerinde çalışır

Bu sayede hem gerçek hayattaki kütüphane işleyişi modellenmiş hem de modern web uygulama mimarisi uygulanmıştır.

2-KULLANILAN TEKNOLOJİLER

Backend

- Python
- Flask
- Flask-JWT-Extended
- Flask-SQLAlchemy
- Flask-Migrate

Veritabanı

- PostgreSQL
- İlişkisel veritabanı yapısı
- Foreign Key ilişkileri

Frontend

- HTML5
- CSS
- JavaScript (Fetch API)

Test Araçları

- **Postman** (API testleri)
- **pgAdmin** (veritabanı kontrolü)

3-SİSTEM MİMARİSİ

Proje **3 katmanlı mimari** ile geliştirilmiştir:

Kullanıcı (Browser)
↓
Frontend (HTML + JS)
↓
Backend (Flask API)
↓
PostgreSQL Veritabanı

Katmanlar arası iletişim:

- Frontend → Backend: `fetch()` + JSON
- Backend → Veritabanı: SQLAlchemy ORM
- Yetkilendirme: JWT Token

4-VERİTABANI TASARIMI

Tablolar

kullanici

Alan	Açıklama
id	Kullanıcı ID
ad	Kullanıcı adı
emai l	Giriş için kullanılan e- posta
sifre	Hashlenmiş parola
rol	admin / user

kitap

Alan	Açıklama
id	Kitap ID
ad	Kitap adı
yazar_id	Yazar ilişkisi
kategori_i d	Kategori ilişkisi
stok	Mevcut stok

odunc

Alan	Açıklama
id	Ödünç ID
kullanici_id	Kitabı alan kullanıcı
kitap_id	Alınan kitap
baslangic_tari h	Ödünç tarihi
iade_tarih	Son iade tarihi
teslim_edildi	Durum bilgisi

ceza

Alan	Açıklama
id	Ceza ID
odunc_id	Hangi ödünçten doğduğu
miktar	Ceza tutarı
tarih	Ceza tarihi
aciklama	Ceza açıklaması

5-KİMLİK DOĞRULAMA ve GÜVENLİK (JWT)

JWT Kullanımı

- Kullanıcı giriş yaptığında `/api/login` endpoint'i çalışır
- Doğru bilgiler girilirse **JWT token** üretilir
- Token frontend'de **localStorage**'da saklanır
- Yetkili endpoint'lerde `@jwt_required()` kullanılır

Avantajları:

- Stateless (sunucuda session tutulmaz)
- Güvenli
- Rol bazlı kontrol yapılabilir

6-ROL BAZLI SİSTEM

Admin Yetkileri

- Kitap ekleme
- Tüm ödünçleri görüntüleme
- Tüm cezaları görüntüleme
- Kullanıcıları listeleme

Kullanıcı Yetkileri

- Kitapları görüntüleme
- Kitap ödünç alma
- Kendi ödünçlerini görme
- Kendi cezalarını görme

Kullanıcılar sadece kendilerine ait verileri görebilir.

7-ÖDÜNÇ ALMA MEKANİZMASI

1. Kullanıcı kitap listesinden kitap seçer
2. `/api/oduncler` → POST isteği gönderilir
3. Stok kontrolü yapılır
4. Stok varsa:
 - Ödünç kaydı oluşturulur
 - Kitap stoğu 1 azaltılır
 - İade tarihi otomatik hesaplanır (7 gün)

8-İADE ve CEZA SİSTEMİ

- **Kullanıcı kitabı iade ettiğinde:**
 - Teslim durumu güncellenir
 - Kitap stoğu artırılır
- **İade tarihi geçmişse:**
 - Gecikme gün sayısı hesaplanır
 - $\text{Gün} \times 5 \text{ TL}$ şeklinde ceza oluşturulur
 - Ceza tablosuna otomatik kayıt eklenir

Bu sistem tamamen backend tarafında otomatik çalışır.

9-FRONTEND SAYFALARI

Login

- Kullanıcı giriş ekranı
- JWT token alınır

User Paneli

- Dashboard (Ödünç sayısı, ceza sayısı)
- Kitaplar
- Ödünçlerim
- Cezalarım

Admin Paneli

- Kitap ekleme
- Genel ödünçler
- Genel cezalar

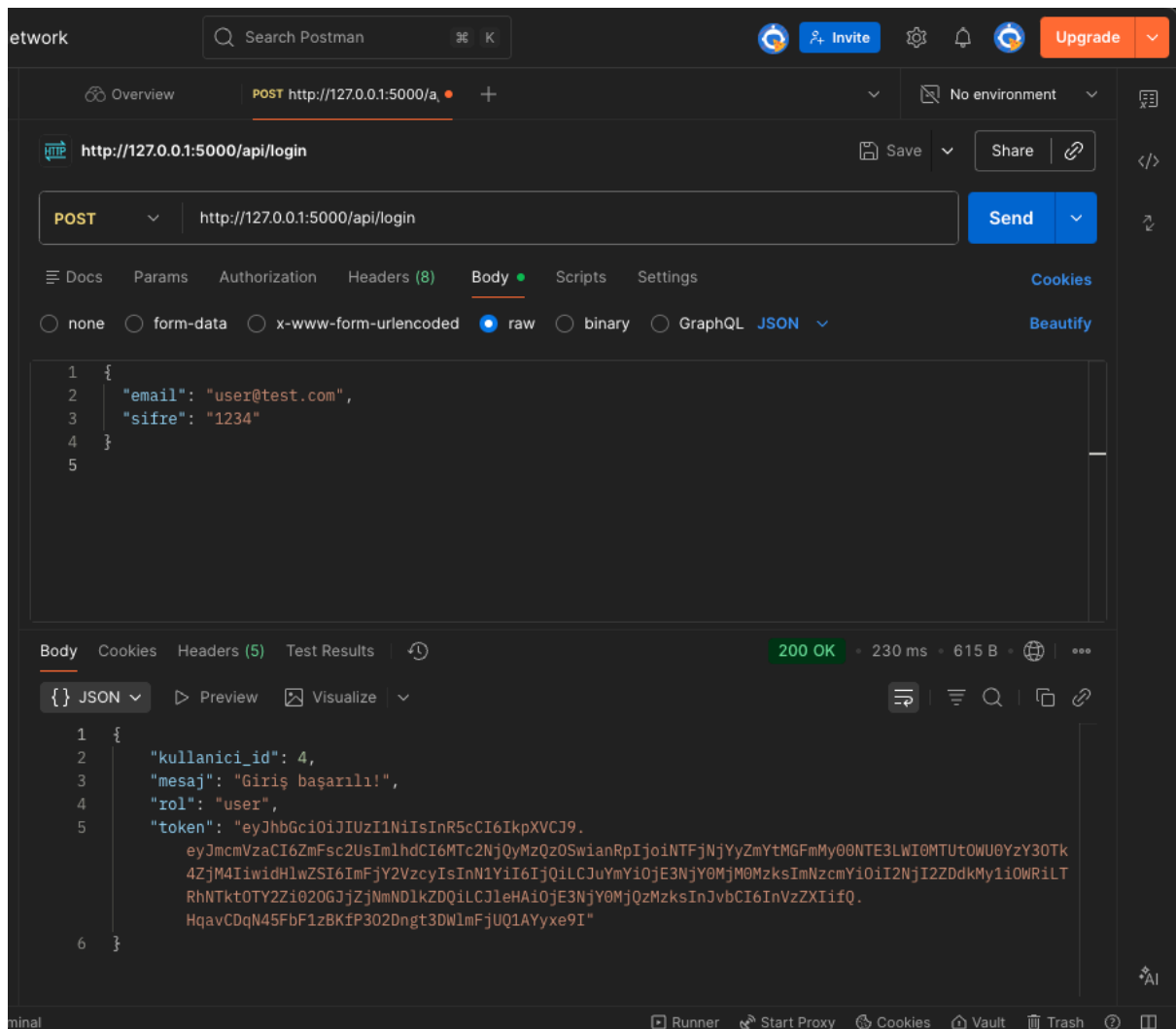
10-API TESTLERİ (POSTMAN)

Projede tüm API uç noktaları **Postman** ile test edilmiştir.

Test edilen başlıca endpoint'ler:

- POST /api/login
- GET /api/kitaplar
- POST /api/oduncler
- GET /api/user/oduncler
- PUT /api/oduncler/{id}/iade
- GET /api/user/cezalar

Tüm test sonuçları Postman Collection olarak kaydedilmiştir ve sunumda gösterilmeye uygundur.



Postman interface showing a GET request to `http://127.0.0.1:5000/api/kitaplar`. The request is successful, returning a 200 OK status. The response body is a JSON array of book objects.

Request: GET `http://127.0.0.1:5000/api/kitaplar`

Authorization: Bearer Token `1zBKfP3O2Dngt3DWlmFjUQ1AYyxe9I`

Response Body (JSON):

```
[
  {
    "ad": "Suç ve Ceza",
    "id": 1,
    "kategori_id": 1,
    "stok": 8,
    "yazar_id": 1
  },
  {
    "ad": "Test Kitap",
    "id": 10,
    "kategori_id": 1,
    "stok": 7,
    "yazar_id": 1
  }
]
```

Postman interface showing a POST request to `http://127.0.0.1:5000/api/oduncler`. The request is successful, returning a 201 CREATED status. The response body is a JSON object indicating the book was borrowed.

Request: POST `http://127.0.0.1:5000/api/oduncler`

Request Body (JSON):

```
{
  "kitap_id": 1
}
```

Response Body (JSON):

```
{
  "mesaj": "Kitap ödünç alındı"
}
```

network Search Postman K

Overview POST http://127.0.0.1:5000/api/user/oduncler GET http://127.0.0.1:5000/api/user/oduncler POST http://127.0.0.1:5000/api/user/oduncler GET http://127.0.0.1:5000/api/user/oduncler + No environment

http://127.0.0.1:5000/api/user/oduncler Save Share

GET http://127.0.0.1:5000/api/user/oduncler Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type Bearer Token Token 1zBKfP3O2Dngt3DWlmFJUQ1AYxe9I

Body Cookies Headers (5) Test Results 200 OK 10 ms 637 B

JSON Preview Visualize

```
1 [
2   {
3     "baslangic_tarih": "2025-12-21",
4     "iade_tarih": "2025-12-28",
5     "id": 12,
6     "kitap_adi": "Suç ve Ceza",
7     "teslim_edildi": false
8   },
9   {
10    "baslangic_tarih": "2025-12-21",
11    "iade_tarih": "2025-12-28",
12    "id": 13,
13    "kitap_adi": "Suç ve Ceza",
14    "teslim_edildi": false
15  },
16  {
17    "baslangic_tarih": "2025-12-22",
18    "iade_tarih": "2025-12-29",
19    "id": 14,
20    "kitap_adi": "Suç ve Ceza",
21    "teslim_edildi": false
22  }
23 ]
```

Runner Start Proxy Cookies Vault Trash ?

network Search Postman K

Overview POST http://127.0.0.1:5000/api/oduncler/14/iade GET http://127.0.0.1:5000/api/oduncler/14/iade POST http://127.0.0.1:5000/api/oduncler/14/iade GET http://127.0.0.1:5000/api/oduncler/14/iade PUT http://127.0.0.1:5000/api/oduncler/14/iade + No environment

http://127.0.0.1:5000/api/oduncler/14/iade Save Share

PUT http://127.0.0.1:5000/api/oduncler/14/iade Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Auth Type Bearer Token Token 1zBKfP3O2Dngt3DWlmFJUQ1AYxe9I

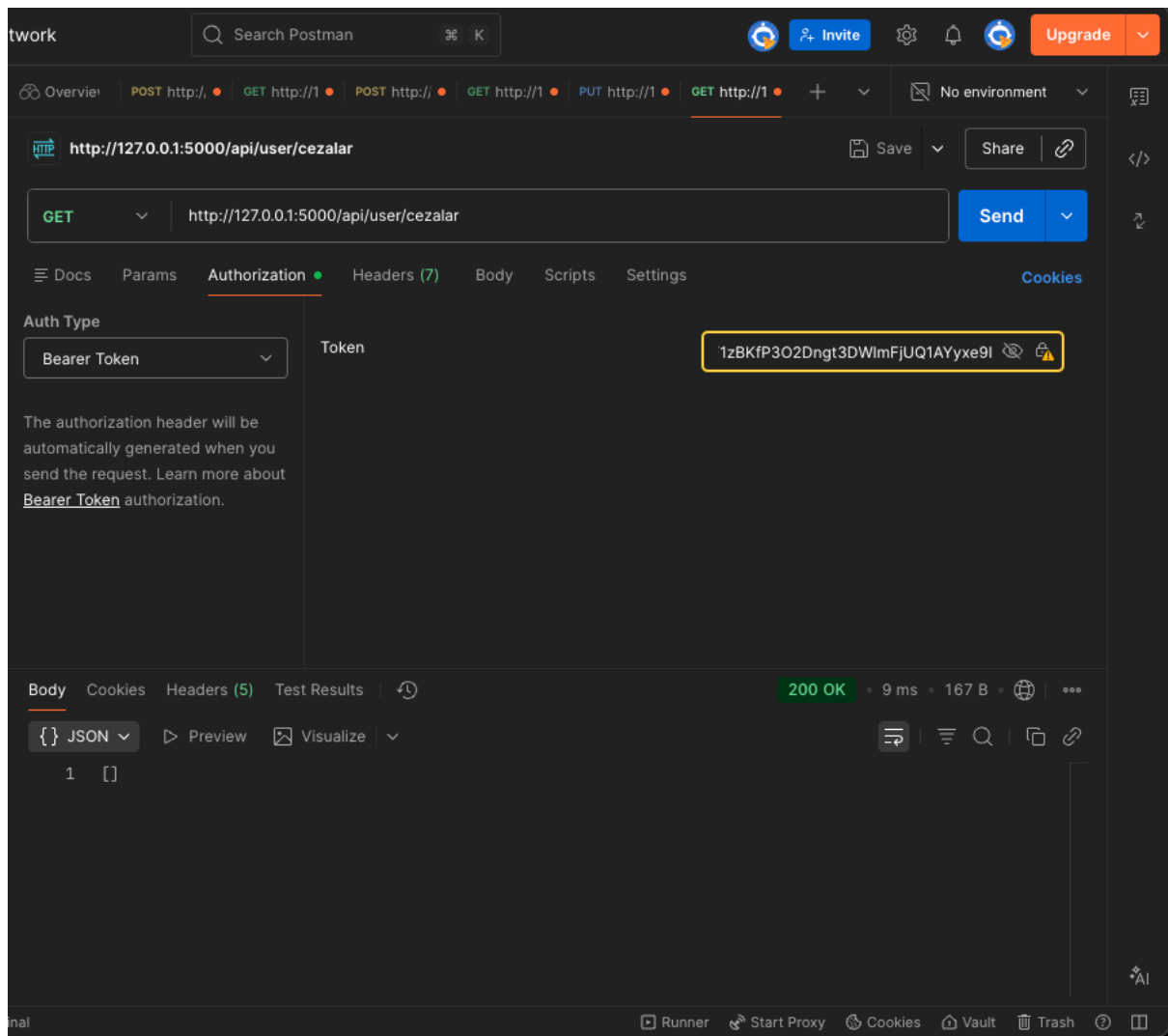
The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (5) Test Results 200 OK 48 ms 200 B

JSON Preview Visualize

```
1 {
2   "mesaj": "Kitap iade edildi"
3 }
```

Runner Start Proxy Cookies Vault Trash ?



```
Terminal Local x + v
(venv) (base) nehirbuketpolat@Nehir-MacBook-Air AkilliKutuphane % python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 168-363-243
127.0.0.1 - - [22/Dec/2025 20:10:39] "POST /api/login HTTP/1.1" 200 -
127.0.0.1 - - [22/Dec/2025 20:12:26] "GET /api/kitaplar HTTP/1.1" 200 -
127.0.0.1 - - [22/Dec/2025 20:14:25] "POST /api/oduncler HTTP/1.1" 201 -
127.0.0.1 - - [22/Dec/2025 20:15:22] "GET /api/user/oduncler HTTP/1.1" 200 -
127.0.0.1 - - [22/Dec/2025 20:16:55] "PUT /api/oduncler/14/iade HTTP/1.1" 200 -
127.0.0.1 - - [22/Dec/2025 20:17:58] "GET /api/user/cezalar HTTP/1.1" 200 -
```

11-PROJENİN KAZANDIRDIKLARI

Bu proje sayesinde:

- Flask ile REST API geliştirme
- JWT ile güvenli kimlik doğrulama
- ORM (SQLAlchemy) kullanımı
- Gerçek hayat senaryosu modelleme
- Frontend–Backend entegrasyonu
- Postman ile profesyonel test süreci

konularında kapsamlı bir uygulama gerçekleştirilmiştir.

12-SONUÇ

Akıllı Kütüphane Yönetim Sistemi, modern yazılım geliştirme prensiplerine uygun, güvenli, modüler ve genişletilebilir bir yapıya sahiptir.

Sistem hem **akademik** hem de **gerçek dünya** ihtiyaçlarını karşılayacak şekilde tasarlanmıştır.