

## **Proje Adı:** Akıllı Kütüphane Yönetim Sistemi

**Teknolojiler:** Flask (Python), Spring Boot (Java), .NET (C#)

**Amaç:** Bu projede öğrencilerin bir kütüphane yönetim sistemi geliştirerek ilişkisel veritabanı tasarımı, temel SQL sorguları (CRUD), veritabanı ilişkileri (JOIN), tetikleyici (TRIGGER) ve prosedür (STORED PROCEDURE) yazımı, REST API geliştirimi, frontend arayüz tasarımı ve backend katmanlı mimari kavramlarını öğrenmeleri hedeflenmektedir.

**Proje Açıklaması:** Öğrenci, bir kütüphane sisteminde yer alan kitapların, kullanıcıların (öğrenciler veya personel), yazarların, kategorilerin ve ödünç verme işlemlerinin yönetilmesini sağlayan bir otomasyon sistemi geliştirecektir. Kullanıcılar sisteme giriş yaparak kitap arayabilecek, ödünç alabilecek ve iade işlemlerini gerçekleştirebilecektir. Geç iade edilen kitaplar için sistem otomatik olarak ceza hesaplayacak, bu ceza bilgisi ayrı bir tabloda saklanacaktır. Admin yetkisine sahip kullanıcılar kitap, yazar ve kategori gibi verileri sisteme ekleyebilecek, güncelleyebilecek ya da silebilecektir. Projede kullanılacak veritabanı sistemi olarak MSSQL, MySQL veya PostgreSQL tercih edilebilir. Uygulamanın backend tarafı ise Flask, Spring Boot veya .NET ile geliştirilecektir. Öğrenci bu üç backend teknolojisinden birini seçerek API'ler oluşturacak ve veritabanı ile bağlantı kurarak gerçek zamanlı işlemleri yönetecektir. Geliştirilen sistemin sadece arka plan (API) kısmı değil, aynı zamanda kullanıcıların etkileşimde bulunacağı bir görsel arayüz (frontend) de içermesi beklenmektedir.

### **Backend:**

- Flask, Spring Boot veya .NET ile API uç noktaları (GET/POST/PUT/DELETE)
- Katmanlı mimari: Entity, Repository, Service, Controller
- JWT tabanlı kimlik doğrulama
- E-posta bildirim sistemi (geç iade)

### **Veritabanı:**

- MSSQL, MySQL veya PostgreSQL tercih edilebilir

### **Frontend:**

- Öğrenci, API'leri kullanan işlevsel bir kullanıcı arayüzü geliştirmelidir
- Frontend teknolojisi olarak HTML/CSS + JavaScript, Thymeleaf (Spring) kullanılabilir
- Giriş (login), kitap arama/listeleme, ödünç alma/iade, admin CRUD gibi temel ekranlar yer almalıdır

### **Test ve Sunum:**

- REST API uç noktaları Postman veya Swagger ile test edilmelidir
- Öğrenci proje testlerini gösteren kısa bir tanıtım videosu çekmeli ve YouTube'a yüklemelidir
- Videoda API'lerin çalıştığı, veri akışının açıklandığı ve örnek senaryoların gösterildiği bir demo sunulmalıdır

**Proje çıktısı:** Tüm işlevsel API'leri çalışan bir backend uygulaması, tercih edilen veritabanı sisteminde tanımlı veri şeması ve fonksiyonlar ile birlikte temel kullanıcı arayüzüne sahip çalışan bir frontend uygulamasıdır.

ER diyagramları, test senaryoları, Swagger veya Postman test koleksiyonu, frontend arayüz tasarımı ve kullanım senaryoları öğrencinin kendi katkısıyla geliştirilecektir.