# Etkinlik Rezervasyon API Case Study

## Genel Bakış

Online bir etkinlik biletleme ve rezervasyon sistemi için REST API geliştirmeniz beklenmektedir. Bu sistem, etkinlik listeleme, koltuk rezervasyonu ve bilet satın alma işlemlerini yönetmelidir.

## Teknik Gereksinimler

### Kullanılacak Teknolojiler

* PHP 8.1+
* Laravel/Symfony Framework
* MySQL/PostgreSQL
* RESTful API standartları
* JWT (JSON Web Token) authentication

### Veritabanı Yapısı

Minimum gereksinimler:

* events (id, name, description, venue\_id, start\_date, end\_date, status, created\_at, updated\_at)
* venues (id, name, address, capacity, created\_at, updated\_at)
* seats (id, venue\_id, section, row, number, status, price, created\_at, updated\_at)
* reservations (id, user\_id, event\_id, status, total\_amount, expires\_at, created\_at, updated\_at)
* reservation\_items (id, reservation\_id, seat\_id, price, created\_at, updated\_at)
* tickets (id, reservation\_id, seat\_id, ticket\_code, status, created\_at, updated\_at)

## API Endpoint Gereksinimleri

### Authentication

* POST /api/auth/register
* POST /api/auth/login
* POST /api/auth/refresh
* POST /api/auth/logout

### Etkinlik İşlemleri

* GET /api/events
* GET /api/events/{id}
* POST /api/events (Admin only)
* PUT /api/events/{id} (Admin only)
* DELETE /api/events/{id} (Admin only)

### Koltuk İşlemleri

* GET /api/events/{id}/seats
* GET /api/venues/{id}/seats
* POST /api/seats/block
* DELETE /api/seats/release

### Rezervasyon İşlemleri

* POST /api/reservations
* GET /api/reservations
* GET /api/reservations/{id}
* POST /api/reservations/{id}/confirm
* DELETE /api/reservations/{id}

### Bilet İşlemleri

* GET /api/tickets
* GET /api/tickets/{id}
* GET /api/tickets/{id}/download
* POST /api/tickets/{id}/transfer

## Önemli İş Kuralları

1. Rezervasyon İşlemleri:
   * Koltuk müsaitlik kontrolü yapılmalı
   * Rezervasyon 15 dakika süreyle geçerli olmalı
   * Aynı koltuk eşzamanlı rezerve edilememeli
   * Rezervasyon süresi dolduğunda koltuklar otomatik serbest bırakılmalı
2. Bilet İşlemleri:
   * Her bilet için benzersiz kod üretilmeli
   * Bilet transferi sadece kullanılmamış biletler için yapılabilmeli
   * Satın alınan biletler PDF formatında indirilebilmeli
   * Bilet iptali etkinlik başlangıç tarihinden 24 saat öncesine kadar yapılabilmeli
3. Validation:
   * Tüm input’lar validate edilmeli
   * Uygun hata mesajları döndürülmeli
   * Request limit uygulanmalı

## Beklenen Özellikler

### Temel Özellikler

* REST API standartlarına uygun endpoint’ler
* JWT tabanlı authentication
* Request/Response logging
* Input validation
* Uygun HTTP status code’ları
* Hata yakalama ve uygun error response’lar

### Bonus Özellikler

* Unit ve Integration testler
* API dokümantasyonu (Swagger/OpenAPI)
* Rate limiting
* Session tabanlı önbellek kullanımı

## Değerlendirme Kriterleri

1. Kod Kalitesi (35%)
   * Clean code prensipleri
   * SOLID prensipleri
   * Design pattern kullanımı
   * Kod organizasyonu ve yapısı
2. Fonksiyonellik (30%)
   * Tüm endpoint’lerin çalışması
   * İş kurallarının doğru uygulanması
   * Concurrency sorunlarının çözümü
   * Hata senaryolarının yönetimi
3. Performans (20%)
   * Database sorgu optimizasyonu
   * N+1 problemi çözümü
   * Basic caching stratejileri
   * Temel yük testi sonuçları
4. Dokümantasyon (15%)
   * README dosyası
   * API dokümantasyonu
   * Kurulum talimatları
   * Kod yorumları ve açıklamaları

## Teslim Şekli

* Projenin GitHub/GitLab repository’si paylaşılmalı
* README dosyasında detaylı kurulum ve kullanım talimatları olmalı
* Postman/Insomnia collection’ı eklenmiş olmalı
* Veritabanı migration ve seeder dosyaları eklenmiş olmalı

## Zaman Sınırı

* Projenin tamamlanması için 7 iş günü verilecektir