

Angular

Bitirme Projesi

Tarih: 26 Ağustos 2025

Proje Konusu: Eğitim Yönetim Sistemi (LMS – Learning Management System)

- Bu projede eğitim platformu geliştirilecek, platform üzerinde **kullanıcılar, kurslar, dersler, yorumlar ve kayıt işlemleri** olacak.
- Backend için **json-server** kullanılacak, frontend tamamen **Angular** ile geliştirilecek.
- Tasarımda özgünsünüz.

Backend (json-server)

Projenin backend tarafı için db.json dosyası oluşturulacak. Aşağıdaki veri yapıları kullanılabilir. Bu içerik örnek olarak oluşturulmuştur. Sizler proje içeriğinize göre düzenleme yapabilirsiniz.

```
{
  "users": [
    { "id": 1, "name": "Ahmet Yılmaz", "email": "ahmet@example.com", "password": "1234", "role": "student" },
    { "id": 2, "name": "Ayşe Demir", "email": "ayse@example.com", "password": "abcd", "role": "instructor" }
  ],
  "courses": [
    { "id": 1, "title": "Angular Başlangıç", "description": "Temel Angular konuları", "instructorId": 2 }
  ],
  "lessons": [
    { "id": 1, "courseId": 1, "title": "Components Giriş", "videoUrl": "https://..." }
  ],
  "enrollments": [
    { "id": 1, "userId": 1, "courseId": 1 }
  ],
  "comments": [
    { "id": 1, "courseId": 1, "userId": 1, "text": "Çok faydalı bir ders!", "date": "2025-08-25" }
  ]
}
```

Angular'da Beklenen Özellikler

1. Kullanıcı Yönetimi

- **Kayıt Ol** ve **Giriş Yap** ekranları.
- Kullanıcı giriş yaptıktan sonra localStorage içinde token benzeri bir veri tutulacak. Token üretimi için (<https://www.npmjs.com/package/json-server-auth>) paketi kullanılabilir, bu bölüm araştırma gerekmektedir.
- Kullanıcının rolü (student veya instructor) sisteme giriş yaparken alınacak.

2. Role Tabanlı Güvenlik (RBAC)

- student: Kurs görüntüleme, kursa kayıt olma, yorum yapma.
- instructor: Kurs ekleme, düzenleme, silme, ders ekleme.
- **Route Guard** ile role tabanlı yönlendirme yapılacaktır.
- Örnek: /admin sayfası sadece instructor rolüne açık olacak.

3. Kurs ve Ders Yönetimi

- Kurslar listelenecek (tüm kullanıcılar görebilecek).
- Kurs detay sayfasında dersler listelenecek.
- Eğitmen (instructor) kurs ekleyebilecek, düzenleyebilecek, silebilecek.
- Eğitmen kursa ders ekleyebilecek.

4. Kurs Katılım (Enrollment)

- Öğrenciler istedikleri kursa kayıt olabilecek.
- Öğrencinin kayıt olduğu kurslar profil sayfasında listelenecek.

5. Yorum Sistemi

- Öğrenciler kurs detay sayfasında yorum yapabilecek.
- Yorumlar kurs detayında listelenecek.

6. UI & UX

- Angular Material veya Bootstrap kullanılmalı.
- Kurs listesi, kurs detayları ve kullanıcı profili modern bir arayüzle sunulmalı.

7. Angular Konularının Kullanımı (Zorunlu)

- Component
- Routing
- Service & Dependency Injection.
- HttpClient ile CRUD İşlemleri.
- Forms Validations
- Route Guards (AuthGuard, RoleGuard).
- LocalStorage kullanımı (auth state).

Teslim Şartları

1. Proje **30 gün içinde** tamamlanmalıdır.
2. Proje **GitHub repo** olarak paylaşılmalıdır.
 - Repo adı: angular-lms-bitirme-projesi veya buna benzer bir isim.
 - README.md dosyası eklenmeli ve **açıklamalı olmalıdır**.
 - Proje açıklaması
 - Kurulum adımları (json-server + Angular)
 - Kullanılan teknolojiler
 - Örnek kullanıcı hesapları (email, password, role)
3. Commit mesajları **açıklayıcı** olmalıdır.
 - Örn: feat: kurs listeleme componenti eklendi
 - fix: login form validation hatası düzeltildi
4. Proje ekran görüntüleri README içinde yer almalıdır.

Değerlendirme Ölçütleri

- **Fonksiyonellik:** Tüm istenilen özellikler çalışıyor mu?
- **Angular Kullanımı:** Component, Service, Routing, Guard gibi yapılar doğru kullanılmış mı?
- **Kod Kalitesi:** Temiz kod, yorum satırları, modüler yapı.
- **UI/UX:** Kullanıcı dostu arayüz, responsive tasarım.
- **GitHub Kullanımı:** Açıklamalı commitler, README dosyası, repo düzeni.
- **Güvenlik:** Role-based access control doğru çalışıyor mu?