

**Departman: Yazılım Mühendisliği**  
**Ders Adı: Yazılım İnşası YZM 201**

## **TO DO LIST UYGULAMASI**

### **PROJE FİNAL RAPORU**

#### **Adlar ve Soyadlar:**

1. BEİBARYS GALYMZHAN
2. ALISHO ODZHIEV

**Danışman Öğretim Üyesi:** Dr. Öğr. Üyesi Ayşe Nur ALTINTAŞ TANKÜL

**Tarih:** aralık 2026 Karabük

## **İÇİNDEKİLER**

1. GİRİŞ
2. PROJENİN GERÇEKLEŞTİRİLMESİ
3. KARŞILAŞILAN SORUNLAR VE ÇÖZÜMLER
4. TEST VE DOĞRULAMA
5. SONUÇ VE DEĞERLENDİRME
6. GELECEK ÇALIŞMALAR
7. KAYNAKÇA

## **1. GİRİŞ**

Bu rapor, Yazılım İnşası dersi kapsamında geliştirilen **To Do List Uygulaması** projesinin final aşamasını kapsamaktadır. Proje, kullanıcıların görevlerini yönetebileceği bir full-stack web uygulamasıdır.

Uygulama, analiz ve tasarım raporlarında belirtilen gereksinimler doğrultusunda gerçekleştirilmiştir. Backend Go dili ile Gin framework kullanılarak geliştirilmiş, frontend Angular ile oluşturulmuştur. Veritabanı olarak PostgreSQL tercih edilmiştir. Kimlik doğrulama JWT ile sağlanmıştır.

Proje, GitHub reposunda paylaşılmıştır: [https://github.com/ufoinz/yaz\\_ins\\_project](https://github.com/ufoinz/yaz_ins_project)

## 2. PROJENİN GERÇEKLEŞTİRİLMESİ

Proje, planlanan mimariye uygun olarak uygulanmıştır:

- **Backend Gerçekleştirme:** Clean Architecture prensiplerine uyularak domain, repository, service ve handler katmanları oluşturuldu. Gin ile REST API endpoint'leri tanımlandı (register, login, tasks CRUD). GORM ile PostgreSQL migration'ları otomatikleştirildi.
- **Frontend Gerçekleştirme:** Angular component'leri ile login/register sayfaları, task list ve modal'lar geliştirildi. HTTP istekleri için services kullanıldı. Responsive tasarım Raleway fontu ve gradyan arka plan ile sağlandı.
- **Entegrasyon:** Frontend backend'e proxy üzerinden bağlandı. JWT token local storage'da saklandı.
- **Çalıştırma:** Backend air ile hot reload, frontend ng serve ile geliştirildi.

## 3. KARŞILAŞILAN SORUNLAR VE ÇÖZÜMLER

- **Sorun 1:** PostgreSQL bağlantı sorunları (port, password, service çalışmama). **Çözüm:** Hizmet manuel başlatıldı, .env dosyası doğru yapılandırıldı, godotenv kütüphanesi eklendi.
- **Sorun 2:** CORS hataları. **Çözüm:** Gin middleware ile CORS ayarlandı.
- **Sorun 3:** UI'de footer overlay sorunu. **Çözüm:** Flexbox ve fixed positioning ile düzeltildi, içerik padding eklendi.
- **Sorun 4:** Veritabanı migration hataları. **Çözüm:** GORM AutoMigrate kullanıldı.

## 4. TEST VE DOĞRULAMA

- **Unit Test:** Backend'de handler'lar için testler yazıldı (go test).
- **Integration Test:** Postman ile API endpoint'leri test edildi (register, login, tasks).
- **Manuel Test:** Tarayıcıda tüm akışlar kontrol edildi (kayıt, giriş, görev CRUD).
- **Kullanıcı Testi:** Farklı cihazlarda responsive davranış doğrulandı.

Sonuç: Tüm fonksiyonel gereksinimler karşılandı.

## 5. SONUÇ VE DEĞERLENDİRME

Proje başarıyla tamamlandı. Kullanıcılar görevlerini güvenli bir şekilde yönetebilmektedir. Clean architecture ve modern teknolojiler sayesinde kod okunaklı ve genişletilebilir hale geldi.

Elde edilen kazanımlar:

- Full-stack geliştirme deneyimi
- REST API tasarımı
- JWT authentication
- Angular component yönetimi

Proje, Yazılım İnşası dersinin hedeflerini karşılamaktadır.

## 6. GELECEK ÇALIŞMALAR

- Görevlere öncelik (priority) ve kategori ekleme
- Bildirim sistemi (email veya push)
- Mobil uygulama (Angular → Ionic)
- Çoklu dil desteği
- Deploy (Heroku/Vercel + Docker)

## KAYNAKÇA

1. Gin Web Framework – <https://gin-gonic.com>
2. GORM – <https://gorm.io>
3. Angular – <https://angular.io>
4. PostgreSQL – <https://www.postgresql.org>
5. JWT – <https://jwt.io>

## EKLER

- Uygulama ekran görüntüleri
- UML Diyagramları (Use Case, Sequence, Class)
- GitHub reposu: [https://github.com/ufoinz/yaz\\_ins\\_project](https://github.com/ufoinz/yaz_ins_project)