



YENİ DERS ÖNERİ FORMU

FAKÜLTE / ENSTİTÜ ADI	Fen-Edebiyat
BÖLÜM / PROGRAM / ANABİLİM DALI ADI	Matematik
DERSİN ADI	Dağıtık Sistemlerde Dayanıklı Uygulamalar Geliştirmek
DERSİN KODU	MTH4050
YEREL KREDİSİ	3
AKTS KREDİSİ	5
HAFTALIK DERS SAATİ	3
HAFTALIK UYGULAMA SAATİ	0
HAFTALIK LABORATUVAR SAATİ	0
ÖNKOŞULLAR	En az bir programlama dili ile uğraşmış ve etkin şekilde kullanıyor olmak (C#, Java, Python, Go vs) güncel yazılım geliştirme platformları (Microsoft .Net, Java Enterprise Edition vb) hakkında bilgi sahibi olmak gerekir.
YARIYIL	Güz
DERSİN DİLİ	İngilizce, Türkçe
DERSİN SEVİYESİ	Lisans
DERSİN TÜRÜ	Seçmeli @Matematik Lisans Programı
DERSİN KATEGORİSİ	Temel Meslek Dersleri
DERSİN VERİLİŞ ŞEKLİ	Yüz Yüze/Çevrimiçi
DERSİ SUNAN AKADEMİK BİRİM	Matematik Bölümü
DERSİN KOORDİNATÖRÜ	Mehmet Emin KÖROĞLU
ASİSTAN(LAR)	
DERSİN AMACI	Bu dersin amacı öğrencilere yüksek dayanıklılığa sahip(Resilient), ölçeklenebilir(Scalable), gözlemlenebilir(Tracable) ve hata toleranslı(Fault Tolerance) dağıtık sistemler(Distributed Systems) geliştirme yetkinliği kazandırmaktır.
DERSİN İÇERİĞİ	Bu ders, yazılım mimarilerinin tarihsel gelişiminden başlayarak öğrencileri mikroservis tabanlı dağıtık sistemlerin temelleri ile tanıştırır. Servisler arası iletişim yöntemleri, yapılandırma ve hata toleransı gibi sistem dayanıklılığını etkileyen başlıca teknikler uygulamalı olarak ele alınır. İzleme (observability), loglama, metrik toplama ve dağıtık izleme gibi araçlarla sistem içgörüsü sağlanır. Sistemlerin kendi kendini onarması, kaos mühendisliği ve rate limiting gibi ileri seviye stratejiler, gerçek senaryolarla desteklenerek işlenir. Final projesinde öğrenciler, tüm bu bileşenleri bir araya getirerek dayanıklı ve gözlemlenebilir bir mikroservis sistemi tasarlar ve test eder.
DERS KİTABI / MALZEMESİ / ÖNERİLEN KAYNAKLAR	1. Kleppmann, Martin. <i>Designing Data-Intensive Applications</i>, O'Reilly Media, 2017. 2. Beyer Betsy, Jones Chris, Petoff Jennifer, Murphy Niall Richard. <i>Site Reliability Engineering: How Google Runs Production Systems</i>, O'Reilly Media, 2016. 3. Tanenbaum Andrew S., Van Steen Maarten. <i>Distributed Systems: Principles and Paradigms (2nd Edition)</i>, Pearson Education, 2007. 4. Abbott Martin L., Fisher Michael T. <i>The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise(2nd Edition)</i>, Addison-Wesley, 2015. 5. Rosenthal Casey, Jones Nora. <i>Chaos Engineering: System Resiliency in Practice</i>, O'Reilly Media, 2020.

**Ders Öğrenim Çıktıları**

1. Yazılım mimarilerinin evrimsel sürecini öğrenir ve dağıtık sistemlerin ortaya çıkış nedenlerini kavrar.
2. Mikroservis mimarisi, servis sınırlandırma (bounded context) ve servis iletişim yöntemleri hakkında uygulamalı bilgi edinir.
3. Sistem dayanıklılığı (resilience), hata toleransı (fault tolerance) ve gözlemlenebilirlik (observability) kavramlarını benimser.
4. Konfigürasyon, kimlik doğrulama, rate limiting ve chaos engineering gibi ileri düzey sistem yönetim tekniklerini uygular.
5. Prometheus, Grafana, Consul, API Gateway, gRPC gibi araç ve protokollerle dağıtık sistemlerin inşasını ve bakımını deneyimler.
6. Farklı diller (.NET, Rust, Go, Python) kullanarak çok katmanlı servislerin işleyişini ve entegrasyonunu yönetebilir.
7. Gerçek hayat senaryolarında hata ayıklama, izleme, loglama ve performans analizlerini sistematik biçimde gerçekleştirme becerisi kazanır.

DEĞERLENDİRME SİSTEMİ

Etkinlikler	Sayı	Katkı Payı
Devam/Katılım		
Laboratuvar		
Uygulama		
Arazi Çalışması		
Derse Özgü Staj		
Küçük Sınavlar/Stüdyo Kritiği		
Ödev		
Sunum/Jüri		
Projeler		
Seminer/Workshop		
Ara Sınavlar	2	60
Final	1	40
Dönem İçi Çalışmaların Başarı Notuna Katkısı		60
Final Sınavının Başarı Notuna Katkısı		40
TOPLAM		100

HAFTALIK KONULAR VE İLGİLİ ÖN HAZIRLIK ÇALIŞMALARI

HAFTALAR	KONULAR	Ön Hazırlık
1	Giriş & Dağıtık Sistemlere Genel Bakış	Kaynak 1 ve Kaynak 3
2	Dağıtık Sistemlere Giriş (<i>Introduction to Distributed Systems</i>)	Kaynak 1 ve Kaynak 3
3	Microservice Mimari ve Bounded Context	Kaynak 1 ve Kaynak 4
4	Servisler Arası İletişim	Kaynak 1
5	Fault Tolerance ve Retry Mekanizmaları	Kaynak 1 ve Kaynak 4
6	Health Check ve Self-Healing Sistemler	Kaynak 1 ve Kaynak 2
7	Konfigürasyon ve Secrets Yönetimi	Kaynak 2 ve Kaynak 4
8	Ara Sınav	
9	Observability (Gözlemlenebilirlik)	Kaynak 2 ve Kaynak 4
10	Resilience Engineering ve Chaos	Kaynak 5
11	Resilience Engineering ve Chaos	Kaynak 5
12	Rate Limiting ve API Gateway Kullanımı	Kaynak 2 ve Kaynak 4
13	Rate Limiting ve API Gateway Kullanımı	Kaynak 2 ve Kaynak 4
14	Eventual Consistency ve Saga Pattern	Kaynak 1 ve Kaynak 4
15	Service Discovery ve Load Balancing	Kaynak 4
16	Final	Kaynak 1 ve Kaynak 3



AKTS İŞYÜKÜ TABLOSU

Etkinlikler	Sayı	Süresi (Saat)	Toplam İşyükü
Ders Saati	14	3	42
Laboratuar			
Uygulama			
Arazi Çalışması			
Sınıf Dışı Ders Çalışması	14	3	42
Derse Özgü Staj			
Ödev			
Küçük Sınavlar/Stüdyo Kritiği			
Projeler			
Sunum / Seminer			
Ara Sınavlar (Sınav Süresi + Sınav Hazırlık Süresi)	2	25	50
Final (Sınav Süresi + Sınav Hazırlık Süresi)	1	20	20
Toplam İşyükü :			154
Toplam İşyükü / 30(s) :			5,13
AKTS Kredisi:			5