

# Mühendislik Öğrencileri İçin Etüt / Ders Arkadaşı Bulma Platformu

## Proje Hedefleri

### 1. Kullanıcı Kaydı ve Girişi

- Öğrenciler ad, e-posta ve şifre ile kayıt olur.
- Kendi profil bilgilerini görüntüleyebilir ve güncelleyebilir.

### 2. Ders Seçimi ve Takibi

- Öğrenciler ilgilendikleri dersleri seçer.
- Sistem, aynı dersi seçmiş diğer öğrencileri kullanıcıya önerir.

### 3. Etüt / Ders Arkadaşı Talep Sistemi

- Kullanıcılar diğer öğrencilere “çalışma arkadaşı olma” talebi gönderir.
- Talep kabul edilirse eşleşme gerçekleşir ve karşılıklı görülebilir.

### 4. Puanlama ve Sosyal Özellikler

- Eşleşme sonrası kullanıcılar birbirini puanlayabilir (1-5 yıldız).
- Favori arkadaş listesi oluşturulabilir.

### 5. Admin Paneli

- Kullanıcı ve ders yönetimi
- Taleplerin ve eşleşmelerin takibi
- Gereksiz veya uygunsuz eşleşmelerin silinmesi

### 6. Görsellik ve Tema

- Hazır HTML tema ile dashboard ve kullanıcı kartları oluşturulacak.
- Her öğrencinin profil fotoğrafı, adı ve seçtiği dersler kart üzerinde gösterilecek.

## Proje Planı

### Hafta Hedefler

<b>Hafta 1</b>	MVC projesinin oluşturulması ve temel proje yapısının kurulması. Controller ve view'ların hazırlanması.
<b>Hafta 2</b>	Kullanıcı kayıt ve giriş sisteminin geliştirilmesi; profil sayfasının oluşturulması ve temel kullanıcı bilgilerini görüntüleme.
<b>Hafta 3</b>	Ders seçimi özelliğinin eklenmesi; kullanıcıların ilgilendiği dersleri seçebilmesi ve seçilen derslerin listelenmesi.
<b>Hafta 4</b>	Talep / eşleşme mantığının geliştirilmesi (talep gönderme, gelen ve giden taleplerin listelenmesi).
<b>Hafta 5</b>	Eşleşmelerin onaylanması / reddedilmesi; eşleşmelerin kullanıcı arayüzünde gösterilmesi.
<b>Hafta 6</b>	Puanlama ve favori arkadaş listesi eklenmesi; kullanıcıların birbirlerini değerlendirebilmesi.
<b>Hafta 7</b>	Hazır HTML tema entegrasyonu; dashboard ve kart tasarımı ile görselliğin artırılması; responsive tasarım kontrolü.
<b>Hafta 8</b>	Projenin genel testlerinin yapılması, hataların düzeltilmesi ve tüm işlevselliğin çalıştığından emin olunması.

### HAFTA 1: MVC Projesinin Oluşturulması ve Temel Yapı

**Haftalık Hedef:** MVC projesinin oluşturulması ve temel proje yapısının kurulması. Controller ve view'ların hazırlanması.

#### Yapılan İşler:

1. Proje Altyapısı: ASP.NET Core 8.0 MVC projesi başlatıldı ve temel dosya yapısı oluşturuldu.
2. Veritabanı Bağlantısı: MongoDB.Driver paketi kuruldu. Program.cs dosyasına, appsettings.json üzerindeki ayarlardan okunan MongoDB Client ve Database servisleri (Dependency Injection) eklendi.

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
courses	-	20.48 kB	4	111.00 B	1	20.48 kB
studyRequests	-	36.86 kB	4	109.00 B	1	36.86 kB
users	-	36.86 kB	5	247.00 B	1	36.86 kB

3. **Temel Veri Modelleri:** Projenin ana veri yapıları olan Models/User.cs ve Models/Course.cs modelleri oluşturuldu.

## HAFTA 2: Kullanıcı Kayıt/Giriş ve Profil Görüntüleme

**Haftalık Hedef:** Kullanıcı kayıt ve giriş sisteminin geliştirilmesi; profil sayfasının oluşturulması ve temel kullanıcı bilgilerini görüntüleme.


### Yapılan İşler:

- Kimlik Doğrulama:** Program.cs'e Cookie Authentication ayarları eklendi. Güvenli şifre saklama için BCrypt.Net-Next paketi kullanılarak şifreler hash'lendi.
- Kayıt/Giriş Akışı:** Controllers/AccountController.cs içinde Register ve Login metotları yazıldı.
- Dinamik Ana Sayfa:** Giriş yapan kullanıcıyı otomatik olarak Kontrol Paneli'ne (Dashboard) yönlendiren HomeController.cs mantığı kuruldu.
- Profil Düzenleme:** Views/Account/Profile.cshtml (Profili Düzenle formu) oluşturuldu ve kullanıcının temel bilgilerini güncelleme altyapısı hazırlandı.

## Etüt Arkadaşım ile çalışmak artık çok kolay.

Mühendislik öğrencileri için ders ve proje arkadaşı bulma platformu. Sizinle aynı dersleri alan öğrencileri keşfedin, eşleşin ve birlikte çalışmaya başlayın.

[Hemen Başla](#)[Giriş Yap](#)

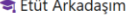
 [Ana Sayfa](#) [Arkadaş Bul](#) [Giriş Yap](#) [Kayıt Ol](#)

## Kayıt Ol

Yeni bir hesap oluşturun.

Kayıt Ol

© 2024 - EtutArkadasim.Web

 [Ana Sayfa](#) [Arkadaş Bul](#) [Giriş Yap](#) [Kayıt Ol](#)

## Giriş Yap

Oturum açmak için bilgilerinizi girin.

Giriş Yap

Hesabınız yok mu? [Kayıt Olun.](#)

© 2024 - EtutArkadasim.Web

### HAFTA 3: Ders Seçimi Özelliği

**Haftalık Hedef:** Ders seçimi özelliğinin eklenmesi; kullanıcıların ilgilendiği dersleri seçebilmesi ve seçilen derslerin listelenmesi.

#### Yapılan İşler:

- API Geliştirme:** Controllers/CoursesController.cs API'si oluşturuldu. Bu API'ler, tüm dersleri getirme (/api/courses/getall) ve kullanıcıya ders ekleme (/api/courses/add/{id}) mantığını içerir.

2. **Dinamik Veri Çekimi:** Views/Home/Dashboard.cshtml dosyasına eklenen JavaScript (Fetch API) kodları, /api/courses/getmycourses adresinden kullanıcının seçtiği dersleri çeker ve Kontrol Paneli'ndeki listeyi dinamik olarak günceller.
3. **Teknik Kanıt:** İstemci tarafında (Frontend) JSON veri transferi kullanılarak API mimarisi kanıtlanmıştır.

Seçtiğim Dersler

+ Ders Ekle

Veri Yapıları (CENG201)

Tüm Dersler

×

Eklemek istediğiniz dersi seçin:

Veri Yapıları (Bilgisayar Müh.)

Diferansiyel Denklemler (Makine Müh.)

Termodinamik (Kimya Müh.)

Devre Analizi (Elektrik-Elektronik Müh.)

Kapat

Seçtiğim Dersler

+ Ders Ekle

Veri Yapıları (CENG201)

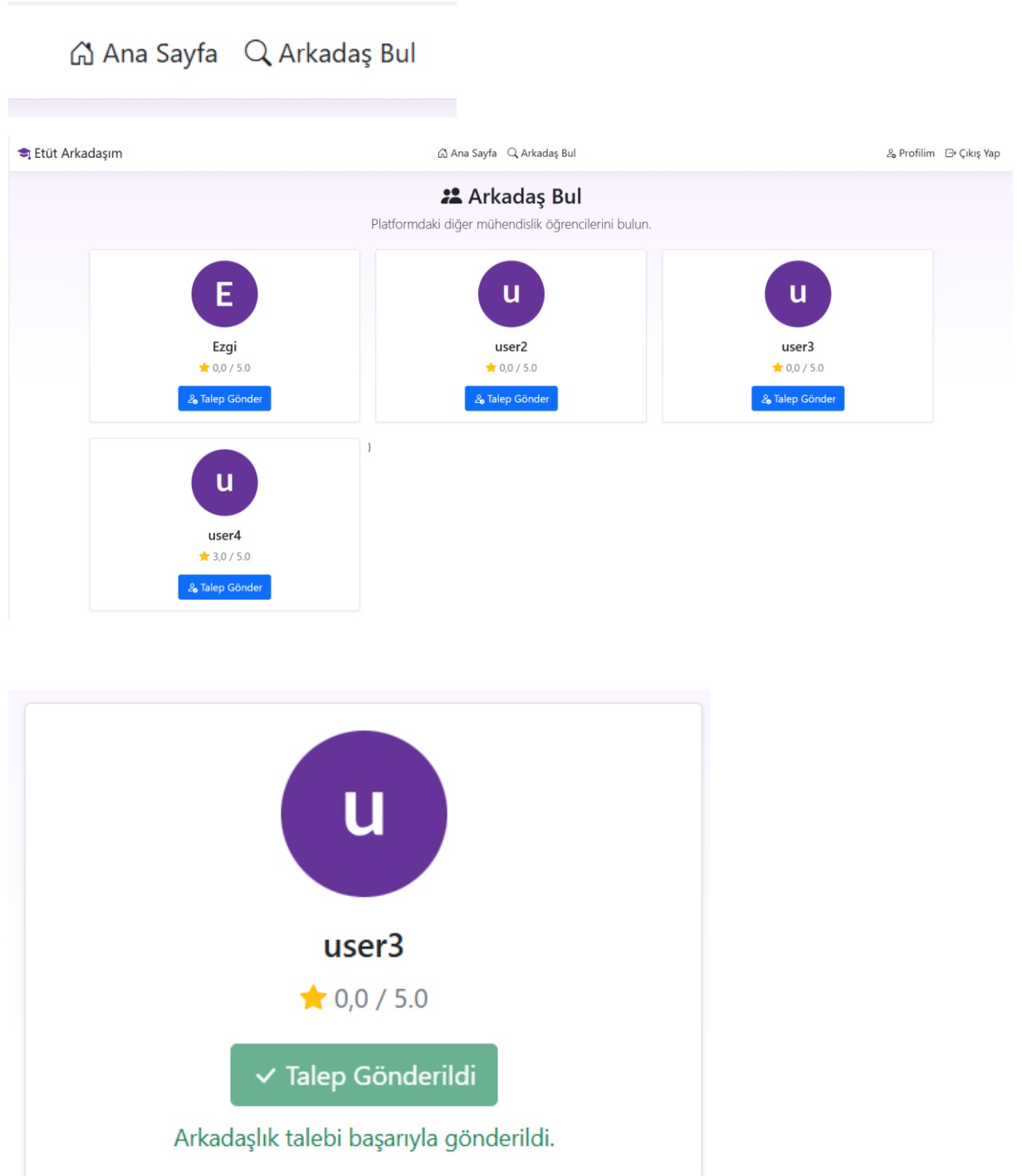
Devre Analizi (EE201)

#### HAFTA 4: Talep Gönderme Sistemi

**Haftalık Hedef:** Talep / eşleşme mantığının geliştirilmesi (talep gönderme, gelen ve giden taleplerin listelenmesi).

**Yapılan İşler:**

1. **Talep Modeli:** Models/StudyRequest.cs (Pending, Accepted, Rejected) modeli oluşturuldu.
2. **Talep API:** Controllers/StudyRequestController.cs oluşturuldu.  
/api/request/send/{receiverId} (POST) metodu ile talep gönderme mantığı yazıldı.
3. **Kullanıcı Bul Sayfası:** Views/Discover/Index.cshtml (Arkadaş Bul) oluşturuldu.  
Kullanıcılar kendileri hariç tüm öğrencileri kartlar halinde görür.
4. **Frontend (Gönderim):** Discover/Index.cshtml'deki JavaScript, butona tıklandığında API'yi çağırır ve anlık başarı/hata mesajı gösterir.





## HAFTA 5: Eşleşmelerin Yönetimi (Gelen Kutusu)

**Haftalık Hedef:** Eşleşmelerin onaylanması / reddedilmesi; eşleşmelerin kullanıcı arayüzünde gösterilmesi.

### Yapılan İşler:

1. **Gelen Kutusu API:** /api/request/pending (GET) ile kullanıcının kendisine gelen beklemedeki talepler listelenir.
2. **Kabul ve Red API:** /api/request/accept/{id} ve reject/{id} (POST) metotları yazıldı. Kabul işlemi, her iki kullanıcının favoriteUserIds listesini karşılıklı güncelleyerek kalıcı eşleşmeyi sağlar.
3. **Arkadaşlarımı Listeleme:** /api/request/myfriends (GET) API'si Dashboard'daki "Arkadaşlarım" kartını doldurur.
4. **Frontend (Gelen Kutusu):** Dashboard'daki "Gelen Etüt Talepleri" kartı, bu API'ları kullanarak talepleri listeler ve "Kabul Et/Reddet" butonlarını dinamik olarak ekler.


 **Profil Bilgileri**




**user3**  
user3@ornek.com

---

**Ortalama Puan:** ★ 0,0 / 5.0

 [Profili Düzenle](#)

 **Gelen Etüt Talepleri**

user

✓ Kabul Et

✗ Reddet

👤 Arkadaşlarım (Eşleşmeler)	
👤 user (★ 5.0)	★ Puan Ver
👤 user4 (★ 3.0)	★ Puan Ver

## HAFTA 6: Puanlama ve Favori Sosyal Özellikler

**Haftalık Hedef:** Puanlama ve favori arkadaş listesi eklenmesi; kullanıcıların birbirlerini değerlendirebilmesi.

### Yapılan İşler:

- Puanlama API:** Controllers/RatingController.cs oluşturuldu ve /api/rating/rate (POST) metodu yazıldı. API, yalnızca arkadaş olunan kişilere puan verilmesini kontrol eder.
- Yıldızlı Arayüz:** Puanlama işlemi, Bootstrap Icons kullanılarak 1-5 yıldız seçimi yapılan, şık bir modal (popup) arayüzü ile tamamlanmıştır. Verilen puan, anında kullanıcının ortalama puanını günceller.
- Profil Güncelleme:** Profil düzenleme özelliği (Hafta 2'de temeli atılan), Ad ve Fotoğraf URL'si güncelleme işlevi ile bitirilmiştir.

👤 Arkadaşlarım (Eşleşmeler)	
👤 user (★ 5.0)	★ Puan Ver
👤 user4 (★ 3.0)	★ Puan Ver

Puan Ver: user

'user' ile çalışmanızı 1-5 arası puanlayın:

★

★

★

★

★

İptal

✓ Puanı Kaydet



## HAFTA 7 & 8: Görsel İyileştirmeler ve Proje Final

**Haftalık Hedef:** Hazır HTML tema entegrasyonu; dashboard ve kart tasarımı ile görselliğin artırılması; responsive tasarım kontrolü ve genel testler.

### Yapılan İşler:

1. **Tema ve Stil:** wwwroot/css/site.css dosyası güncellenerek mor temaya uygun (Mor Tema, Mor-Beyaz Gradient Arka Plan) modern bir stil seti uygulandı.
2. **Görsel Tasarım:** Tüm navigasyon ve kart tasarımları Bootstrap Icons ile zenginleştirildi. Ana sayfa, profesyonel bir karşılama (Hero Section) tasarımı ile güncellendi.
3. **Sonuç:** Proje, mobil uyumlu (responsive) ve tüm işlevselliği çalışan bir duruma getirilmiştir.
4. **Genel Testler:** Tüm API'lar (Swagger) ve Frontend etkileşimleri (Talep/Kabul/Puanlama) test edilmiştir.

## Projedeki API Uç Noktalarının Kullanımı

Projenin API'lerinin (CoursesController, StudyRequestController, RatingController) doğru çalıştığını kanıtlamak amacıyla **Swagger UI** entegrasyonu yapılmıştır.

Courses		^
GET	/api/Courses/addsamples	▼
GET	/api/Courses/getall	▼
GET	/api/Courses/getmycourses	▼
POST	/api/Courses/add/{courseId}	▼
Rating		^
POST	/api/rating/rate	▼
StudyRequest		^
POST	/api/request/send/{receiverId}	▼
GET	/api/request/pending	▼
POST	/api/request/accept/{requestId}	▼
POST	/api/request/reject/{requestId}	▼
GET	/api/request/myfriends	▼

## 1. CoursesController (Ders Yönetimi - Hafta 3)

Bu grup, ders listelerini yönetir ve kullanıcıların ders seçimi yapmasını sağlar.

- **GET /api/courses/getall**
  - **Görevi:** Veritabanındaki tüm derslerin listesini JSON formatında döndürür.
  - **Kullanım Yeri:** Dashboard'daki "Ders Ekle" modalının içini doldurmak için kullanılır.
- **GET /api/courses/getmycourses**
  - **Görevi:** Giriş yapmış kullanıcının seçtiği tüm dersleri (ID'ler üzerinden) çekip döndürür.
  - **Kullanım Yeri:** Dashboard'daki "Seçtiğim Dersler" kartını dinamik olarak doldurur.
- **POST /api/courses/add/{courseId}**
  - **Görevi:** Belirtilen ders ID'sini, giriş yapmış kullanıcının selectedCourseIds listesine ekler.
  - **Kullanım Yeri:** "Ders Ekle" modalında bir derse tıklandığında çağrılır.

## 2. StudyRequestController (Talep ve Eşleşme Yönetimi - Hafta 4 & 5)

Bu grup, sosyal etkileşimi ve arkadaşlık durumunu yönetir.

- **POST /api/request/send/{receiverId}**
  - **Görevi:** Giriş yapmış kullanıcıdan, belirtilen alıcı ID'sine yeni bir etüt arkadaşı talebi (status: Pending) oluşturur.
  - **Kullanım Yeri:** "Arkadaş Bul" (Discover) sayfasındaki "Talep Gönder" butonuna basıldığında kullanılır.
- **GET /api/request/pending**
  - **Görevi:** Giriş yapmış kullanıcıya gelen, durumu "Beklemede" (Pending) olan tüm talepleri, gönderen kullanıcının adıyla birlikte listeler.
  - **Kullanım Yeri:** Dashboard'daki "Gelen Etüt Talepleri" kartını doldurur.
- **POST /api/request/accept/{requestId}**
  - **Görevi:** Belirtilen talep ID'sinin durumunu "Kabul Edildi" yapar ve iki kullanıcıyı da birbirinin favoriteUserIds listesine ekleyerek eşleşmeyi tamamlar.
  - **Kullanım Yeri:** Dashboard'daki "Kabul Et" butonuna basıldığında çağrılır.

- **GET /api/request/myfriends**

- **Görevi:** Giriş yapmış kullanıcının eşleştiği (favori) tüm arkadaşları puan bilgileriyle birlikte listeler.
- **Kullanım Yeri:** Dashboard'daki "Arkadaşlarım" kartını doldurur.

### 3. RatingController (Puanlama Yönetimi - Hafta 6)

Bu grup, kullanıcıların birbirini değerlendirme işlevini sağlar.

- **POST /api/rating/rate**

- **Görevi:** Gönderilen puanı (Score), puanlanan arkadaşın (RatedUserId) ratings listesine ekler ve ortalama puanı günceller. Sadece eşleşilen kişilere puan verilebilir.
- **Kullanım Yeri:** Dashboard'daki "Puan Ver" modalından, yıldız seçimi yapıp kaydetme butonuna basıldığında çağrılır.

## GELECEK GELİŞTİRME PLANI

### 1. Kullanıcı Hesap Yönetimi ve Veri Akışı İyileştirmeleri

- **Gelişmiş Profil Güncelleme ve Veri Bütünlüğü:** Profil düzenleme sayfası, kullanıcının ad, e-posta ve şifre bilgilerini (mevcut şifre doğrulaması ile) güvenli bir şekilde güncelleyebileceği eksiksiz bir form haline getirilecektir. Bu işlem sırasında tüm veri akışı API üzerinden sağlanacaktır.
- **Fotoğraf Yükleme Mekanizması:** Profil fotoğrafı ataması, URL girmek yerine, kullanıcıların yerel cihazlarından dosya seçerek yükleyebileceği bir sisteme dönüştürülecektir. Bu, sunucu tarafında dosya yönetimi API'sinin tasarımını gerektirecektir.
- **Arayüz Ortalaması:** Kullanıcı giriş ve kayıt formları, tüm ekran tiplerinde estetik ve kullanım kolaylığı için sayfanın tam ortasına hizalanacaktır.

### 2. Ders Yönetimi ve Eşleşme Bütünlüğü

- **Dinamik Ders Yönetimi:**
  - Platformdaki ders seçenekleri artırılacaktır.
  - Kullanıcıların eklediği dersleri kolayca görebileceği ve listedeki dersleri silme işlemini gerçekleştirebileceği dinamik arayüz ve bunu destekleyen API uç noktaları geliştirilecektir.

- **Eşleşme Çakışması Kontrolü:** O an eşleşmiş olan kişilerin, aynı dersi seçen diğer kişilerle yeni eşleşme talebi gönderememesi/alamaması kuralı, StudyRequestController içinde API seviyesinde uygulanacaktır.

### 3. Canlı İletişim ve İlişki Sonlandırma

- **Canlı İletişim Sistemi (Chat):** Eşleşmiş kullanıcıların birbirleriyle anlık olarak iletişim kurabileceği bir sistem tasarlanacaktır.
- **İlişki Yönetimi (Çıkarma ve Sonlandırma):**
  - **Arkadaşlıktan Çıkarma:** Favori arkadaş listesinden (eşleşmelerden) bir kişinin tek taraflı olarak çıkarılabilmesi için API desteği eklenecektir.
  - **Eşleşme Sonlandırma:** Ders çalışmaları sona erdiğinde, kişilerin eşleşmelerini kalıcı olarak sonlandırabileceği bir mekanizma tasarlanacaktır. Bu, gelecekteki yeni eşleşmelere yer açmak için ilgili kayıtların veritabanından temizlenmesini içerecektir.