



**T.C**  
**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ**

**PROJE KONUSU: WEB ARAYÜZÜNDEN OTOMATİK DERS**  
**PROGRAMI OLUŞTURMA**

**HAZIRLAYANLAR**

**AHMET EREN ŞENGÜL 220502036**

**BAŞAR BURAK ÜNAL 220501025**

**YUSUF USTAOęLU 220502003**

**EFE ÖZTAŞ 220501014**

**ONUR KOCA 240502040**

**DERS SORUMLUSU:**  
**Dr. Öğr. Üyesi ELİF PINAR HACİBEYOęLU**

**TARİH: 7.05.2025**

# 1 GİRİŞ

## 1.1 Projenin amacı

- Bu projenin amacı, üniversitelerde ve eğitim kurumlarında ders programlarını yönetmek ve optimize etmek için bir yazılım geliştirmektir. Geliştirilecek yazılım, derslerin, öğretim üyelerinin, dersliklerin ve öğrenci gruplarının yönetimini kolaylaştırmayı hedefler. Aynı zamanda, kullanıcıların zaman ve kaynak verimliliğini artırarak eğitim süreçlerinin daha düzenli ve verimli olmasını sağlamayı amaçlar.
- Proje kapsamında geliştirilen web uygulaması, mühendislik fakültesi dersleri için haftalık ders programlarını otomatik olarak oluşturacaktır. Kullanıcılar, web arayüzü üzerinden ders programlarını görüntüleyebilir, ders seçimleri yapabilir, öğretim üyeleri için haftalık programları görüntüleyebilir ve gerekirse manuel düzenlemeler gerçekleştirebilir. Uygulama, veritabanından alınan bilgilerle otomatik olarak ders programını oluşturacak ve bu sayede kullanıcılar tüm dersler hakkında detaylı bilgilere ulaşabilecektir.

## 1.2 Projede Gerçekleştirilmesi Beklenenler

- Haftalık ders programlarının otomatik olarak oluşturulması
- Kullanıcıların ders seçimlerini yapabilmesi
- Öğretim üyelerinin programlarını görüntüleyebilmesi
- Manuel ders programı düzenlemeleri yapılabilmesi
- Ders programındaki çakışmaların tespiti ve yönetilmesi
- Kullanıcı dostu bir web arayüzü tasarlanması
- Ders programının Excel formatında dışa aktarılabilmesi
- Veritabanı entegrasyonu ile ders, öğretim üyesi ve derslik bilgilerine kolay erişim sağlanması

## 2 GEREKSİNİM ANALİZİ

### 2.1 Arayüz gereksinimleri

#### Kullanıcı Arayüzü Gereksinimleri

- **Kullanıcı Dostu Tasarım:** Arayüz, her seviyeden kullanıcı tarafından kolayca anlaşılır ve kullanılabilir yapıda olmalı; işlemler hızlı ve basit şekilde tamamlanmalı.
- **Duyarlı Tasarım (Responsive Design):** Mobil, tablet ve masaüstü cihazlarla tam uyumlu çalışmalı.
- **Renk ve Kontrast Kullanımı:** Erişilebilirlik standartlarına uygun renk paletleri ve kontrast oranları kullanılmalı.
- **Görsel Takvim:** Ders programı, gün ve saat bazında tablo şeklinde görsel olarak sunulmalı.
- **Ders Seçim ve Kayıt Ekranı:** Öğrenciler dersleri seçip detaylarını görüntüleyerek kayıt işlemini arayüz üzerinden gerçekleştirmeli.
- **Çakışma Tespit ve Yönetim Aracı:** Zamanı çakışan dersler otomatik tespit edilmeli ve kullanıcıya uyarı gösterilmeli.
- **Program Yönetimi Ekranı:** Öğretim üyeleri ve koordinatörler, ders programlarını görüntüleyip düzenleme işlemlerini kullanıcı dostu ekranlar üzerinden yapmalı.

#### Donanım Arayüzü Gereksinimleri

- **Bilgisayarlar ve Mobil Cihazlar:** Uygulama tarayıcı tabanlıdır. Kullanıcıların sisteme erişebilmesi için internet bağlantısına sahip bir cihaz yeterlidir; özel bir donanım gerekmez.

### 2.2 Fonksiyonel gereksinimler

#### Ders Programı Yönetimi

- **Otomatik Ders Programı Oluşturma:** Sistem, veritabanındaki verileri kullanarak çakışmasız haftalık program oluşturmali.
- **Ders Programı Görüntüleme:** Kullanıcı, kendi ders programını gün ve saat bazında görsel olarak görüntüleyebilmeli.
- **Ders Seçme:** Öğrenci, dersi seçtiğinde sistem, dersi otomatik olarak programa eklemeli.
- **Ders Seçim Filtreleme:** Dersler; bölüm, öğretim üyesi ve saat gibi kriterlere göre filtrelenebilmeli.

#### Ders Seçim ve Kayıt

- **Öğrenci Kayıt İşlemleri:** Öğrenciler, kendilerine uygun dersleri seçip kaydedebilir. Seçim sırasında dersin öğretim üyesi, saat ve derslik bilgisi gibi detaylar gösterilir.

- **Ders Kayıt Çakışması:** Derslerin zamanları çakıştığında sistem, kullanıcıyı uyarak çakışmanın çözülmesi için önerilerde bulunur.
- **Kapasite Kontrolü:** Her dersin öğrenci kapasitesi sınırlandırılmıştır. Öğrenci, kapasiteyi aşan bir ders seçmeye çalıştığında sistem uyarı verir.

### **Kullanıcı Yönetimi**

- **Kullanıcı Hesapları:** Öğrenciler, öğretim üyeleri ve program koordinatörleri farklı yetkilerle sisteme giriş yapacaktır. Her kullanıcı tipi sadece yetkili olduğu alanlara erişebilir.
- **Hesap Yönetimi:** Kullanıcılar, hesap bilgilerini güncelleyebilir, şifre değiştirebilir ve sistem üzerinden kendi ders programlarını görüntüleyebilirler.

### **Derslik ve Zaman Yönetimi**

- **Derslik Yönetimi:** Dersler, dersliklerin kapasitesine göre atanacaktır. Derslik kapasitesine göre, her dersin öğrenci sayısı kontrol edilir.
- **Zaman Çakışması Algoritması:** Derslerin zamanları çakıştığında sistem, çakışmayı tespit eder ve kullanıcıya çözüm önerileri sunar.

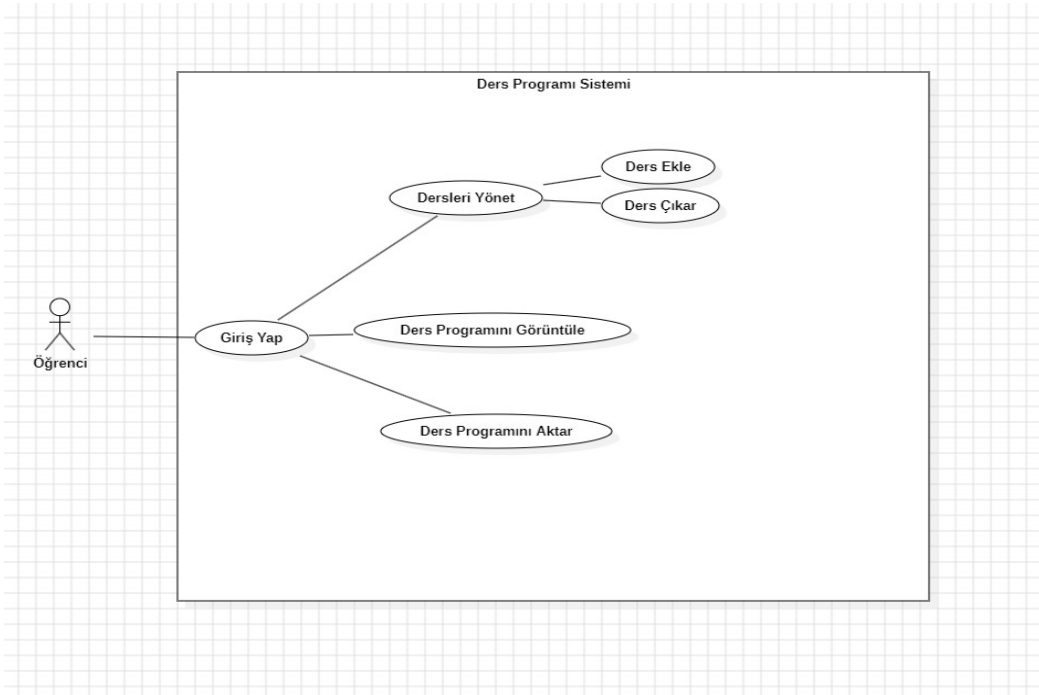
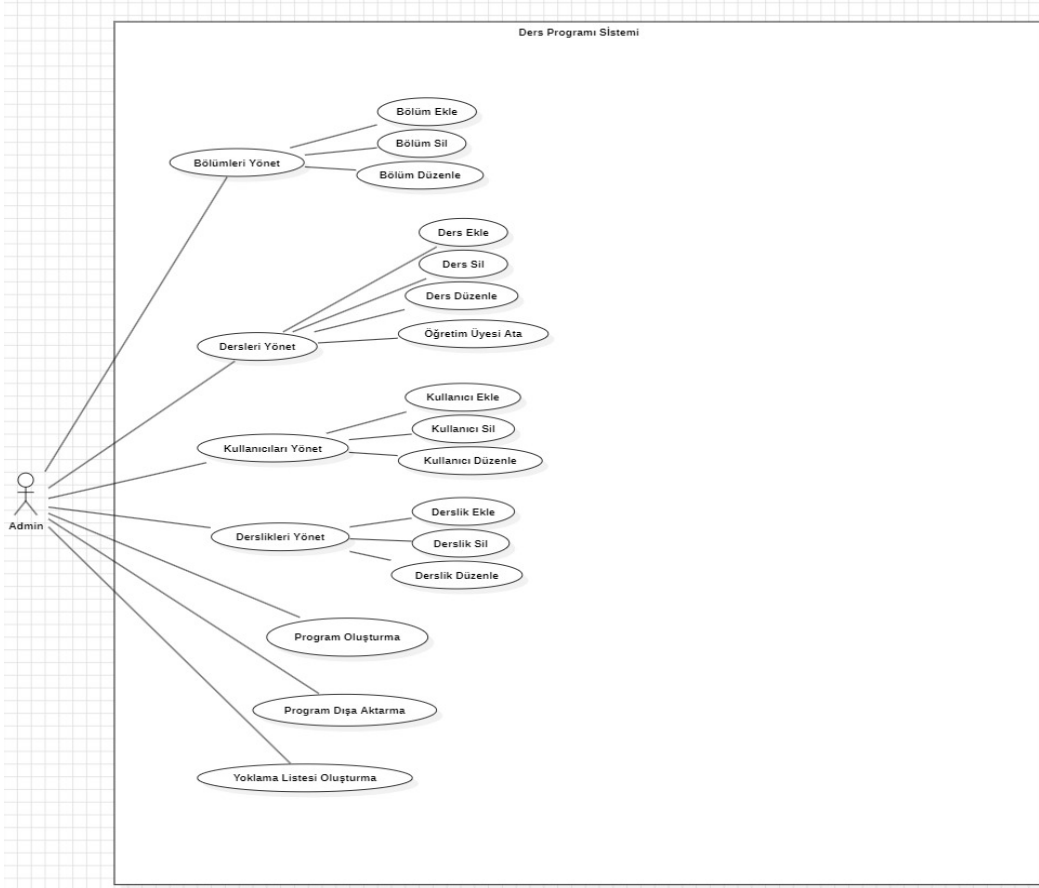
### **Veritabanı Yönetimi**

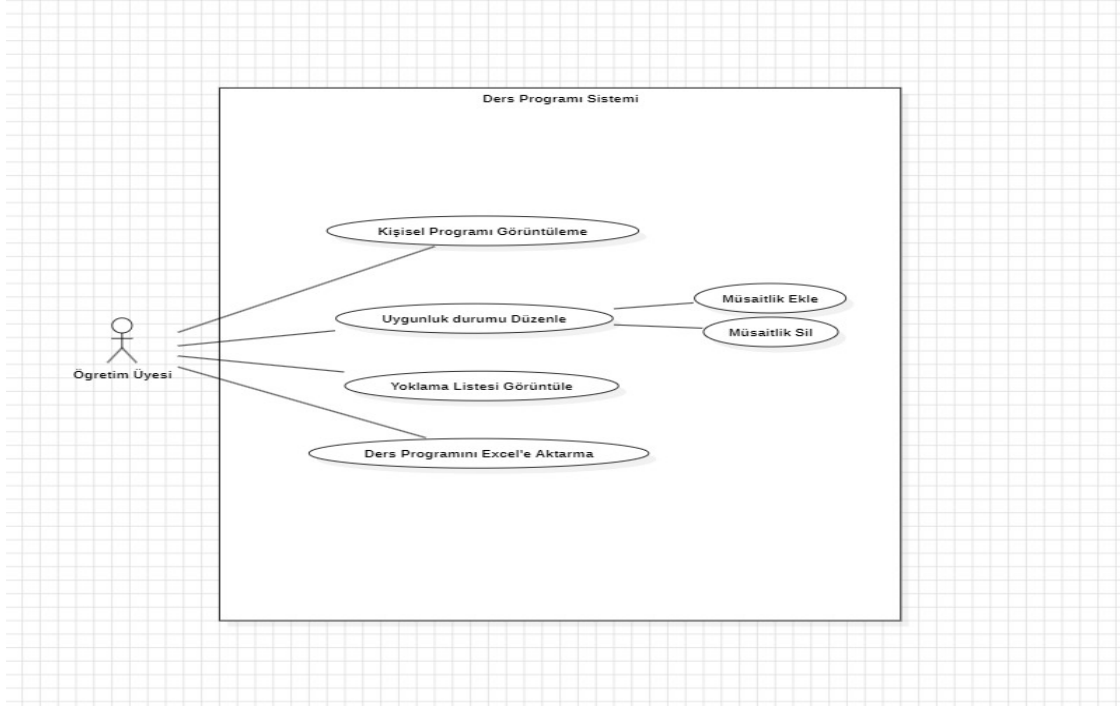
- **Veri Ekleme ve Güncelleme:** Sistem, dersler, öğretim üyeleri, derslikler ve öğrenciler hakkında verileri güncelleyebilecek ve bu verileri veritabanında tutacaktır.

### **Çakışma Yönetimi**

- **Çakışma Tespiti:** Sistem, derslerin öğretim üyeleri ve dersliklerle olan zaman çakışmalarını tespit eder.
- **Çakışma Çözme:** Kullanıcıya çözüm seçenekleri sunulur, gerektiğinde manuel düzenleme yapılır.

## 2.3 Use-Case diyagramı

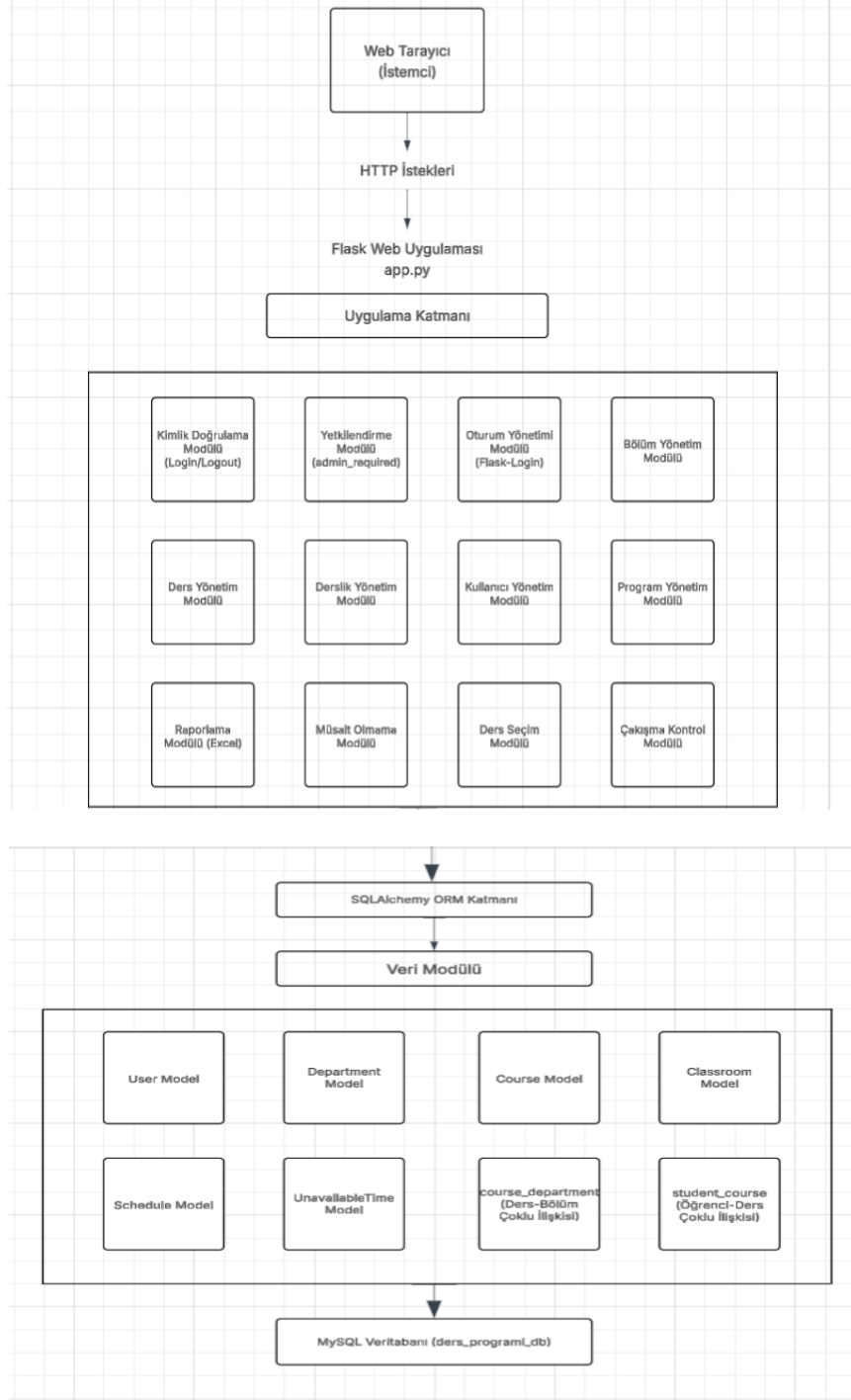




## 3 TASARIM

### 3.1 Mimari tasarım

- Veri Katmanı: MySQL veritabanı ile veri depolama ve erişim
- İş Mantığı Katmanı: Python Flask uygulamasında uygulama mantığının yönetimi
- Sunum Katmanı: HTML, CSS ve JavaScript kullanılarak oluşturulan web arayüzü
- Kullanıcı Yönetimi: Kullanıcı kimlik doğrulama ve yetkilendirme
- Veri Yönetimi: Bölüm, derslik, ders ve program verilerinin yönetimi
- Çakışma Kontrolü: Programda çakışmaları önleyen kontrol mekanizmaları
- Raporlama: Excel formatında dışa aktarma işlemleri



### 3.2 Kullanılacak teknolojiler

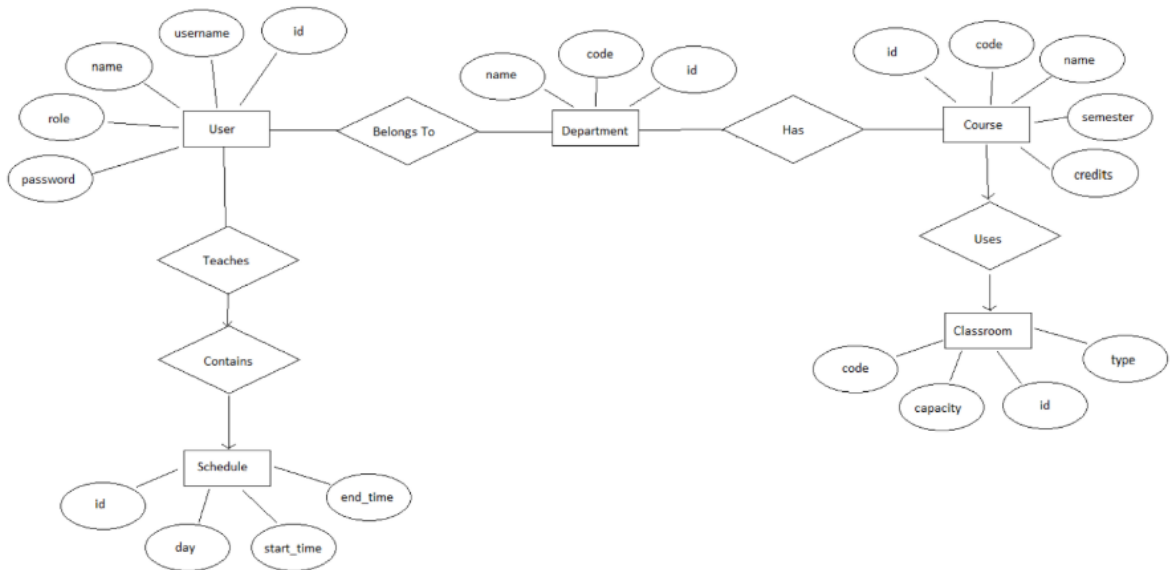
- Programlama Dili: Python
- Web Framework: Flask
- ORM: SQLAlchemy
- Kullanıcı Arayüzü: HTML5, CSS3, JavaScript, Bootstrap 5
- Kullanıcı Kimlik Doğrulama: Flask-Login
- Veritabanı: MySQL
- Arayüz: HTML, CSS, JavaScript, Bootstrap
- Excel İşlemleri: OpenPyXL kütüphanesi
- Flask: Web uygulaması geliştirme
- Flask-Login: Kullanıcı oturum yönetimi
- Flask-SQLAlchemy: Veritabanı ORM işlemleri
- OpenPyXL: Excel dosyalarının oluşturulması ve düzenlenmesi
- Werkzeug: Güvenli parola hash'leme

### 3.3 Veri tabanı tasarımı

- Projede MySQL veritabanı kullanılarak aşağıdaki tablolar tasarlanmıştır:
- **Users (Kullanıcılar) Tablosu:**
  - Kullanıcı bilgilerini saklar
  - Kullanıcı rolleri: admin, öğretim üyesi, öğrenci
  - Öğrenciler için yarıyıl ve öğrenci numarası bilgileri
  - Bölüm bilgisi (foreign key ile departments tablosuna bağlı)
- **Departments (Bölümler) Tablosu:**
  - Bölüm kodları ve isimleri
  - BLM ve YZM bölümleri için özel kayıtlar
  - Bölüm-ders ilişkileri için referans
- **Courses (Dersler) Tablosu:**
  - Ders kodu ve ismi
  - Teorik ve uygulama saatleri
  - Kredi bilgisi
  - Öğretim üyesi ataması
  - Yarıyıl bilgisi
  - Zorunlu/seçmeli durumu
  - Ders türü (yüzyüze/online)
  - Kapasite bilgisi
- **Classrooms (Derslikler) Tablosu:**
  - Derslik kodu
  - Kapasite bilgisi
  - Derslik türü (normal/laboratuvar)



- **Schedules (Ders Programı) Tablosu:**
- Ders-derslik eşleştirmesi
- Gün ve saat bilgileri
- Çakışma kontrolü için kullanılan zaman dilimleri
- **UnavailableTimes (Müsait Olmama Durumları) Tablosu:**
- Öğretim üyelerinin müsait olmadığı zamanlar
- Gün ve saat bilgileri
- Neden bilgisi
- İlişki Tabloları:
- **course\_department:**
- Derslerin birden fazla bölüme ait olabilmesi için
- Many-to-many ilişki
- **student\_course:**
- Öğrencilerin ders seçimleri
- Yarıyıl ve durum bilgisi
- Many-to-many ilişki
- Bu veritabanı tasarımı, projenin tüm gereksinimlerini karşılayacak şekilde yapılandırılmıştır. Özellikle: Çoklu bölüm desteği, Esnek ders programı oluşturma, Kullanıcı rol yönetimi, Çakışma kontrolü, Kapasite yönetimi gibi temel işlevler için gerekli tüm veri yapıları oluşturulmuştur.
- ER Diyagramı



### 3.4 Kullanıcı arayüzü tasarımı

- **Ana Özellikler:**
- Responsive tasarım (mobil uyumlu)
- Bootstrap tabanlı modern arayüz
- Kullanıcı rolüne göre özelleştirilmiş menüler
- Tablo formatında ders programı görüntüleme
- Excel'e aktarma özelliği
- **Uygulama Çalıştırma:**
- Python ve gerekli kütüphanelerin kurulumu
- Veritabanı bağlantısının yapılandırılması
- Flask uygulamasının başlatılması
- Web tarayıcısından localhost:5000 adresine erişim



Please log in to access this page.

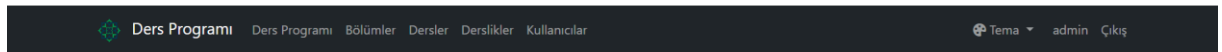
**Giriş Yap**

Kullanıcı Adı

Şifre

**Giriş Yap**

(Giriş ekranı)



Güz dönemi için ders programı başarıyla oluşturuldu.

#### Ders Programı

Otomatik Program Oluştur Excel'e Aktar

**Program Ekle**

Ders

Ders Seçin

Derslik

Derslik Seçin

Gün

Gün Seçin

Başlangıç

--:--

Bitiş

--:--

**Ekle**

#### Bilgisayar Mühendisliği (BLM) Ders Programı

Gün/Saat	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
Pazartesi		<b>BLM213</b> Veritabanı Yönetim Sistemleri 13:00-15:50 D302	<b>BLM315</b> Yazılım Lab-I 09:00-11:50 Online	<b>BLM411</b> Programlama Dilleri 09:00-11:50 M101 <b>BLM413</b> Biyimsel Diller ve Otomatlar 13:00-15:50 S202

<b>Salı</b>	<b>MAT125</b> Lineer Cebir 09:00-11:50 S101  <b>BLM111</b> Bilgisayar Mühendisliğine Giriş 13:00-15:50 D403	<b>MAT211</b> Diferansiyel Denklemler 09:00-11:50 D102  <b>BLM215</b> Temel Elektronik ve Uygulamaları 13:00-15:50 B1LLAB.2	<b>MAT310</b> Sayısal Yöntemler 13:00-15:50 S101	
<b>Çarşamba</b>	<b>FIZ110</b> Fizik I 09:00-11:50 Online  <b>MAT110</b> Matematik I 13:00-15:50 KÜÇÜK LAB	<b>MAT213</b> Ayrık Matematik 13:00-15:50 S201	<b>MAT220</b> Sayısal Yöntemler 09:00-11:50 AMFİ B  <b>BLM311</b> Bilgisayar Mimarisi ve Organizasyonu 13:00-15:50 D104	<b>BLM415</b> Araştırma Problemleri 13:00-15:50 D103
<b>Perşembe</b>	<b>BLM113</b> Bilgisayar Lab I 09:00-11:50 B1LLAB 1	<b>BLM211</b> Nesneye Yönelik Programlama 13:00-15:50 Online	<b>BLM424</b> Yapay Zeka 13:00-15:50 D301 isott	
<b>Cuma</b>	<b>BLM112</b> Bilgisayar Programlama I 09:00-11:50 D104  <b>DİL101</b> İngilizce 13:00-15:50 D403	<b>BLM217</b> Programlama Lab-I 13:00-15:50 B1LLAB.2	<b>BLM313</b> Yazılım Mühendisliği 09:00-11:50 M101  <b>BLM317</b> Yapay Zeka 13:00-15:50 D301	<b>BLM417</b> İş Sağlığı ve Güvenliği-I 09:00-11:50 M301

(Otomatik ders programı oluşturma)

(Program oluştururken dönem seçimi)

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

- Projede geliştirilen yazılım, Flask web framework'ü kullanılarak modüler bir yapıda tasarlanmıştır. Ana bileşenler şu şekilde organize edilmiştir:
- **A. Veritabanı Modelleri (models.py)**
- Veritabanı modelleri, SQLAlchemy ORM kullanılarak oluşturulmuştur. Bu modeller, veritabanı tablolarını Python sınıfları olarak temsil eder ve veri yönetimini kolaylaştırır.
- **User Modeli:**
- Kullanıcı bilgilerini yönetmek için tasarlanmıştır. Kullanıcı rolleri (admin, öğretim üyesi, öğrenci), kimlik bilgileri ve bölüm bilgilerini içerir. Şifre güvenliği için hashleme mekanizması kullanılmıştır.
- **Department Modeli:**
- Bölüm bilgilerini saklar. Her bölümün benzersiz bir kodu ve ismi bulunur. Derslerle many-to-many ilişki kurulmuştur.
- **Course Modeli:**
- Ders bilgilerini yönetir. Ders kodu, ismi, kredisi, kapasitesi gibi temel bilgilerin yanı sıra öğretim üyesi ataması ve yarıyıl bilgisi de tutulur.
- **Classroom Modeli:**
- Derslik bilgilerini içerir. Kapasite ve derslik türü (normal/laboratuvar) bilgileri saklanır.
- **Schedule Modeli:**
- Ders programı kayıtlarını tutar. Ders, derslik ve zaman bilgilerini ilişkilendirir.
- **B. Ana Uygulama Modülü (app.py)**
- Flask uygulamasının ana modülüdür. Tüm route'lar ve iş mantığı burada tanımlanmıştır.
- **Uygulama Konfigürasyonu:**
- Flask uygulaması başlatılır
- Veritabanı bağlantısı yapılandırılır
- Güvenlik ayarları yapılır
- Route Tanımlamaları:
- Kullanıcı girişi ve yetkilendirme
- Ders programı görüntüleme ve düzenleme
- Otomatik program oluşturma
- Excel'e aktarma işlemleri
- İş Mantığı Fonksiyonları:
- Program oluşturma algoritması
- Çakışma kontrolü

- Kapasite yönetimi
- Veri doğrulama
- **C. Şablonlar (templates/)**
- Kullanıcı arayüzü için HTML şablonları Bootstrap framework'ü kullanılarak tasarlanmıştır.
- **Ana Şablonlar:**
- base.html: Tüm sayfalar için temel yapı
- login.html: Kullanıcı girişi
- view\_schedule.html: Ders programı görüntüleme
- **Yönetim Şablonları:**
- departments.html: Bölüm yönetimi
- courses.html: Ders yönetimi
- classrooms.html: Derslik yönetimi
- users.html: Kullanıcı yönetimi
- **Özel Şablonlar:**
- edit\_course.html: Ders düzenleme
- my\_schedule.html: Kişisel program görüntüleme
- alternative\_times.html: Alternatif zaman önerileri

- **D. Yardımcı Modüller**
- **Excel İşlemleri:**
- Program dışı aktarma
- Yoklama listesi oluşturma
- Raporlama
- **Güvenlik:**
- Kullanıcı kimlik doğrulama
- Rol bazlı yetkilendirme
- Şifre hashleme
- **Veri Doğrulama:**
- Form validasyonu
- Çakışma kontrolü
- Kapasite kontrolü
- **Bu modüler yapı sayesinde:**
- Kod tekrarı minimize edilmiştir
- Bakım ve güncelleme kolaylaştırılmıştır
- Yeni özellikler eklemek basitleştirilmiştir
- Hata ayıklama süreci iyileştirilmiştir

## 4.2 Görev dağılımı

- Ekip olarak, haftalık ders programı projesinin her bir aşamasında birlikte ilerledik.
- Rapor ekip üyeleri tarafından iletişimi koparmadan birlikte hazırlanmıştır.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- **Çakışma Kontrolü:** Ders programı oluşturulurken oluşabilecek çakışmaların kontrolü algoritma açısından zorlu bir problemdi. Zaman dilimlerinin kesişimini kontrol eden bir algoritma geliştirilerek çözüldü.
- **İlişkiler Arası Geçiş:** Bir dersin birden fazla bölüme ait olabilmesi için Many-to-Many ilişkisi kullanıldı, bu da kod karmaşıklığını artırdı.
- **Excel Formatı:** Yoklama listesi için özel Excel formatının oluşturulması için openpyxl kütüphanesi ile detaylı stil ayarları yapılması gerekti.
- **Kullanıcı Roller:** Farklı rollerin farklı yetkilere sahip olması için özel decorator fonksiyonlar geliştirildi.

## 4.4 Proje isterlerine göre eksik yönler

- Eksik yön bulunmamaktadır.

# 5 TEST VE DOĞRULAMA

## 5.1 Yazılımın test süreci

- Yazılım test sürecinde testler proje süresi boyunca manuel olarak yapılmıştır.
- Herhangi bir test kodu geliştirilmemiştir.

## 5.2 Yazılımın doğrulanması

- Test Kullanıcı giriş sistemi sorunsuz çalışmaktadır.
- Rol bazlı yetkilendirme sistemi (admin, öğrenci, öğretim üyesi) doğru şekilde işlemektedir.
- Yetkisiz erişim denemeleri başarılı bir şekilde engellenmektedir.
- Bölüm ekleme işlevi düzgün çalışmaktadır.
- Bölüm silme işlemi sorunsuz gerçekleşmektedir.
- Silinen bölümlerin veritabanından tamamen kaldırıldığı doğrulanmıştır.
- Ders ekleme işlevi hatasız çalışmaktadır.
- Ders düzenleme işlemleri eksiksiz gerçekleştirilebilmektedir.
- Derslere öğretim üyesi atama süreci doğru şekilde çalışmaktadır.

- Derslik ekleme işlemi sorunsuz tamamlanmaktadır.
- Derslik düzenleme ve kapasite güncelleme işlemleri başarılı bir şekilde gerçekleştirilmektedir.
- Aynı derslikte aynı saat diliminde farklı derslerin eklenmesi engellenmektedir.
- Aynı öğretim üyesinin aynı saatte farklı derslere atanması önlenmektedir.
- Excel dosyası oluşturma ve indirme işlevi sorunsuz çalışmaktadır.

## 6 KAYNAKÇA

<https://flask-login.readthedocs.io>

<https://docs.sqlalchemy.org/>

<https://www.mysql.com/>

<https://openpyxl.readthedocs.io/en/stable/>

## 7 GitHub LİNKLERİ

Onur Koca <https://github.com/OnurKoca1>

Ahmet Eren Şengül <https://github.com/Eren1213>

Yusuf Ustaoglu <https://github.com/katlicia>

Efe Öztaş <https://github.com/EOztas>

Başar Burak Ünal <https://github.com/BasarBurakUnal>