amblem, logo, simge, sembol, ticari marka içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**T.C.  
KOCAELİ ÜNİVERSİTESİ  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü**

**PROGRAMLAMA VE YAZILIM LABORATUVARI  
SINAV PROGRAMI YÖNETİM SİSTEMİ**

**ÜNİVERSİTE SINAV PROGRAMI HAZIRLAMA UYGULAMASI  
Proje Raporu**

**Yapanlar:  
Engincan Koç 230501011**

**Batuhan Uzuner 230501020**

**Selim İşkodra 23501023**

**Ders Sorumlusu:**

**DR.ÖĞR. ÜYESİ Elif Pınar Hacıbeyoğlu**

**Tarih: 14.01.2026**

**1. GİRİŞ**

**1.1. Projenin Amacı**

Bu projenin temel amacı, bir üniversitedeki sınav planlama sürecini insan hatasından arındırarak **akıllı, kısıt tabanlı ve otomatik** hale getirmektir. Geleneksel yöntemlerle hazırlanan sınav programlarında sıkça karşılaşılan öğrenci sınav çakışmaları, yetersiz derslik kapasitesi ve öğretim görevlilerinin müsaitlik durumları gibi karmaşık değişkenleri merkezi bir sistemde toplar. Algoritma desteğiyle, hem fiziksel kaynakların (derslikler) en verimli şekilde kullanılmasını sağlar hem de öğrencilerin ve hocaların akademik takvimlerini optimize eder.

**1.2. Projede Gerçekleştirilenler**

**Otomatik Sınav Planlama:** Belirlenen tarih aralığı ve çalışma saatleri içerisinde, dersleri öğrenci yoğunluğuna göre sıralayarak en uygun zaman dilimlerine yerleştirmek.

**Çakışma Kontrolü:** Excel'den aktarılan gerçek öğrenci listelerini kullanarak, aynı öğrencinin aynı saatte birden fazla sınavının olmasını engellemek.

**Dinamik Derslik Yönetimi:** Tek bir dersliğin yetmediği durumlarda, derslikler arası yakınlık matrisini kullanarak birbirine en yakın ve uygun kapasitedeki sınıfları otomatik olarak birleştirmek.

**Rol Bazlı Yetkilendirme:** Sistemde Admin (yönetici), Bölüm Yetkilisi, Öğretim Üyesi ve Öğrenci rolleri tanımlayarak, her kullanıcının sadece kendi yetki alanındaki verileri (sınav takvimi, ders ekleme vb.) görmesini sağlamak.

**Veri Entegrasyonu:** Karmaşık öğrenci ve derslik verilerini manuel girmek yerine, standart Excel formatındaki dosyalardan sisteme hızlıca aktarmak.

**Esnek Süre Yönetimi:** 30 dakikadan başlayıp 8 saate (480 dakika) kadar uzanabilen farklı sınav türleri ve süreleri için (ara sınav, final, uygulama) destek sunmak.

**Erişilebilirlik ve Raporlama:** Hazırlanan sınav programlarını profesyonel PDF ve Excel formatlarında dışa aktararak ilan edilmeye hazır hale getirmek.

**Kullanıcı Deneyimi:** Modern web teknolojileri (Sticky nav, responsive tasarım) ile hem bilgisayardan hem de mobil cihazlardan sınav takvimine kolay erişim sağlamak.

**2. GEREKSİNİM ANALİZİ**

**2.1. Arayüz Gereksinimleri**

**Kullanıcı Arayüzü (UI) Gereksinimleri**

Kullanıcıların sistemle etkileşime girdiği görsel ve işlevsel tüm bileşenler modern web standartlarına uygun şekilde tasarlanmıştır.

**Responsive (Duyarlı) Tasarım:** Sistem, farklı ekran boyutlarına (mobil, tablet, masaüstü) tam uyumlu bir yapıda olmalı; ekran daraldığında kartlar, butonlar ve tablolar okunabilirlik kaybı yaşamadan yeniden şekillenmelidir**.**

**Rol Bazlı Navigasyon Menüsü:** Kullanıcının rolüne (Admin, Bölüm Yetkilisi, Hoca, Öğrenci) göre menü içerikleri dinamik olarak değişmeli; sadece yetkili olunan modüllere erişim sağlanmalıdır.

**Sticky (Yapışkan) Gezinme Çubuğu:** Sayfa aşağı kaydırıldığında erişilebilirliği artırmak adına "Sticky Nav" bileşeni devreye girmeli ve hızlı erişim menüsü sunmalıdır.

**Görsel Geri Bildirim ve Animasyonlar:** İşlem sonuçları (başarı, hata, uyarı) için renk kodlu flash mesajlar kullanılmalı; butonlarda "Ripple" efekti ve kartlarda "Fade-in" animasyonları ile kullanıcı etkileşimi güçlendirilmelidir.

**Veri Filtreleme ve Arama Arayüzü:** Sınav programı listesinde fakülte, bölüm ve gün bazlı filtreleme yapabilmeyi sağlayan interaktif seçim araçları bulunmalıdır.

**Erişilebilirlik Desteği:** Yüksek kontrastlı mod desteği ve ekran okuyucular için "sr-only" sınıfları gibi erişilebilirlik standartları karşılanmalıdır**.**

**İnteraktif Formlar:** Veri giriş alanlarında gerçek zamanlı doğrulama (validation) ve odaklanma (focus) efektleri ile hata payı minimize edilmelidir**.**

**Profesyonel Çıktı Arayüzü:** Hazırlanan programların PDF ve Excel formatında indirilebilmesi için görünür ve kullanımı kolay dışa aktarma butonları yer almalıdır.

**Donanım Arayüzü Gereksinimleri**

Proje bir web uygulaması olduğu için doğrudan özel bir donanım üretimi gerektirmez; ancak sistemin kararlı çalışması için aşağıdaki donanım etkileşimleri temel alınmıştır:

* **Sunucu Tarafı (Server-Side):** Uygulamanın ve MySQL veritabanının barındırılması için minimum çift çekirdekli işlemci ve 4 GB RAM kapasitesine sahip bir sunucu donanımı gereklidir.
* **İstemci Cihazlar:** Modern web tarayıcılarını (Chrome, Firefox, Safari vb.) çalıştırabilen; klavye, fare veya dokunmatik ekran girişine sahip standart bilgisayar veya mobil cihazlar yeterlidir.
* **Ağ Bağlantısı:** Sunucu ve istemci arasındaki veri transferi (özellikle Excel import işlemleri sırasında) için kesintisiz internet veya yerel ağ (LAN) bağlantısı bulunmalıdır**.**
* **Yazıcı Arayüzü (Opsiyonel):** Tarayıcı üzerinden PDF çıktılarını fiziksel kağıda dökebilmek için standart yazdırma protokollerine uyumlu donanım desteği**.**

**2.2. Fonksiyonel Gereksinimler**

Sistemin temel işlevlerini ve kullanıcıların gerçekleştirebileceği operasyonları kapsayan gereksinimler şunlardır:

**1. Kullanıcı Yönetimi ve Güvenlik**

* **Kimlik Doğrulama:** Kullanıcılar, kullanıcı adı ve şifreleri ile sisteme giriş yapabilmeli ve oturumlarını sonlandırabilmelidir.
* **Rol Bazlı Yetkilendirme:** Sistem; Admin, Bölüm Yetkilisi, Hoca ve Öğrenci rollerine göre farklı erişim seviyeleri sunmalıdır.
* **Şifre Güvenliği:** Kullanıcı şifreleri veritabanında güvenli bir şekilde (hashlenmiş olarak) saklanmalıdır.

**2. Akademik Veri Yönetimi**

* **Ders Yönetimi:** Yetkili kullanıcılar ders ekleyebilmeli, ders bilgilerini (kod, isim, fakülte, bölüm, hoca, öğrenci sayısı) güncelleyebilmeli veya silebilmelidir.
* **Derslik Yönetimi:** Dersliklerin isim, bina, kat, oda tipi ve sınav yapılabilirliği gibi özellikleri sisteme tanımlanabilmelidir.
* **Hoca Müsaitlik Yönetimi:** Öğretim üyelerinin hangi gün ve saatlerde sınav yapmaya müsait oldukları sisteme girilebilmelidir.

**3. Otomatik Sınav Planlama (Scheduler)**

* **Otomatik Çizelgeleme:** Sistem; ders, derslik ve hoca verilerini kullanarak 10 günlük bir sınav programını otomatik olarak oluşturmalıdır.
* **Çoklu Derslik Ataması:** Tek bir dersliğin kapasitesinin yetmediği durumlarda, sistem birden fazla dersliği bir sınav için atayabilmelidir.
* **Yakınlık Optimizasyonu:** Çoklu derslik atamalarında, fiziksel olarak birbirine en yakın derslikler (yakınlık matrisi verilerine göre) tercih edilmelidir.
* **Çakışma Kontrolü:** Aynı öğrencinin veya aynı dersliğin aynı zaman diliminde birden fazla sınavda bulunması engellenmelidir.
* **Hoca Müsaitlik Uyumu:** Sınavlar, ilgili hocanın sadece sistemde tanımlı olan müsaitlik saatlerine göre planlanmalıdır.

**4. Veri Entegrasyonu ve Dışa Aktarma**

* **Excel'den Veri Aktarımı:** Sınıf listeleri, derslik kapasiteleri ve derslik yakınlık ilişkileri Excel dosyaları üzerinden toplu olarak sisteme aktarılabilmelidir.
* **PDF ve CSV Çıktısı:** Oluşturulan sınav programları, filtrelenmiş veya tam liste olarak profesyonel PDF ve Excel (CSV) formatlarında indirilebilmelidir.
* **Kişisel Takvim Görüntüleme:** Hoca ve öğrenciler, sadece kendi sorumlu oldukları veya kayıtlı oldukları derslerin sınavlarını görebilecekleri özel bir takvime erişebilmelidir.

**USE CASE DİYAGRAMI**

**metin, diyagram, ekran görüntüsü, tasarım içeren bir resim

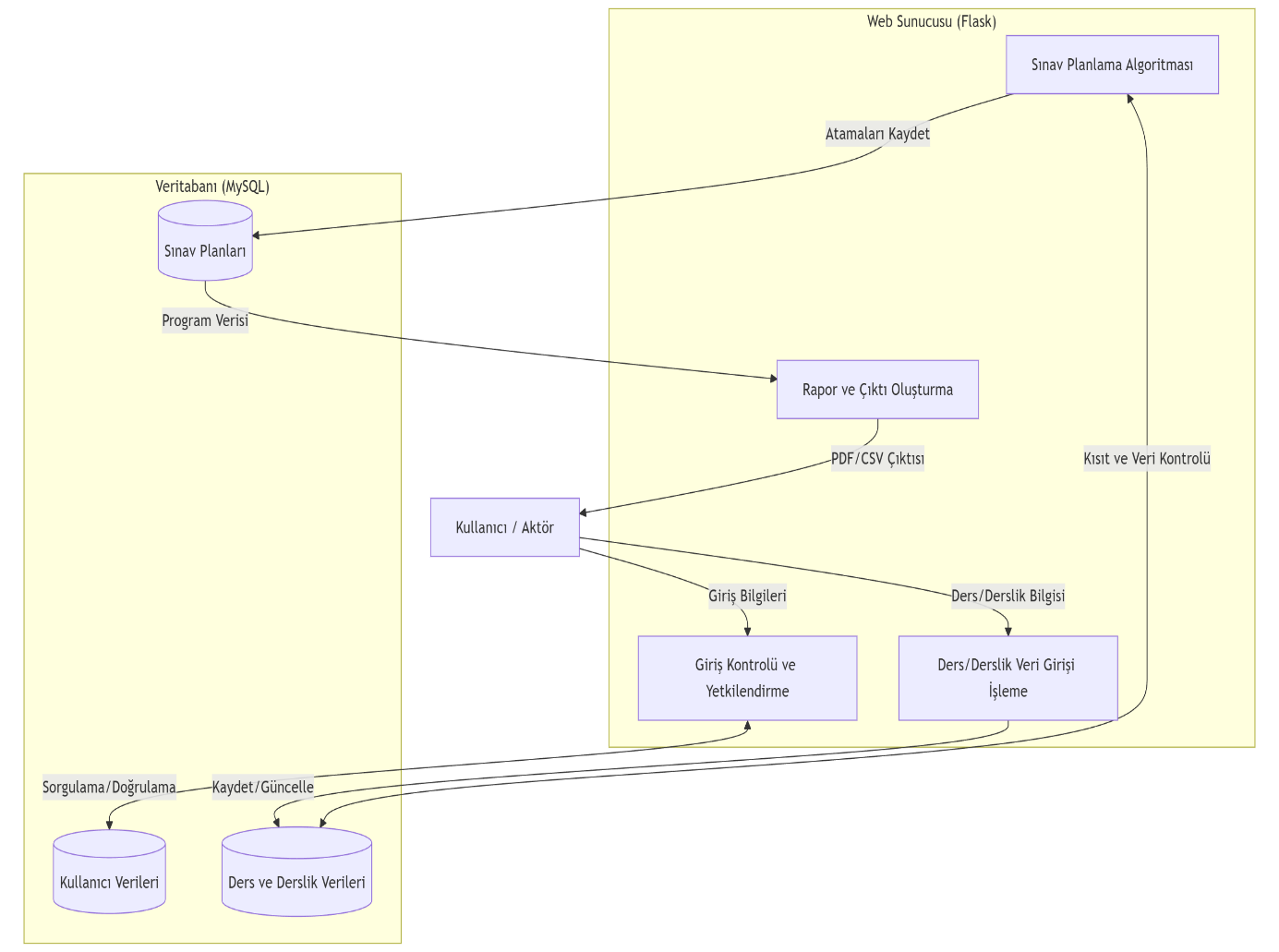
Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**CLASS DİYAGRAMI**

**metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**VERİ AKIŞ DİYAGRAMI**

****

**ER DİYAGRAMI**

**metin, diyagram, paralel, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**3. TASARIM**

**3.1. Mimari Tasarım**

Sistem, modern web uygulaması standartlarına uygun olarak **MVC (Model-View-Controller)** benzeri bir mimari üzerine inşa edilmiştir. Katmanlı yapının detayları şunlardır:

* **Model Katmanı:** Veritabanı tablolarını (Ders, Derslik, Sınav, Kullanıcı vb.) ve bu tablolar arasındaki ilişkileri (Bire-Çok, Çok-Çok) temsil eden models.py dosyasından oluşur.
* **Controller (Yönlendirme) Katmanı:** İstekleri karşılayan, iş mantığını yürüten ve kullanıcıyı ilgili sayfalara yönlendiren routes.py dosyasını kapsar.
* **İş Mantığı ve Algoritma Katmanı:** Sistemin temel işlevlerini gerçekleştiren scheduler.py (planlama algoritması) ve excel\_importer.py (veri aktarımı) modüllerinden oluşur.
* **Görünüm (View) Katmanı:** Kullanıcıya sunulan arayüzlerin (HTML) bulunduğu Jinja2 şablon motoru ile desteklenen klasördür.

**3.2. Kullanılacak Teknolojiler**

**Yazılım Dili:**

* Uygulamanın backend (arka uç) geliştirmesi tamamen **Python** dili ile yapılmıştır.

**Harici Kütüphaneler:**

* **Flask:** Uygulamanın web sunucusu ve rotasyon (routing) yapısını kurmak için kullanılan hafif web çatısıdır.
* **Flask-SQLAlchemy:** Python nesneleri ile MySQL veritabanı arasında köprü kuran (ORM) kütüphanedir.
* **Pandas:** Excel dosyalarından veri okuma ve veri işleme süreçlerini yönetmek için kullanılmıştır.
* **ReportLab:** Sınav programını profesyonel PDF formatında oluşturmak ve dışa aktarmak için entegre edilmiştir.
* **Werkzeug:** Güvenli şifre hashleme ve kullanıcı doğrulama işlemleri için tercih edilmiştir.

**Diğer Teknolojiler:**

* **MySQL:** Tüm akademik verilerin ve sınav kayıtlarının tutulduğu ilişkisel veritabanı yönetim sistemidir.
* **Bootstrap 5:** Arayüzün modern, responsive (mobil uyumlu) ve kullanıcı dostu olması için kullanılan CSS framework'üdür.

**3.3 Kullanıcı Arayüzü Tasarımı**

Arayüz tasarımı, kullanıcı deneyimini (UX) ön planda tutan modern ve temiz bir yapıya sahiptir:

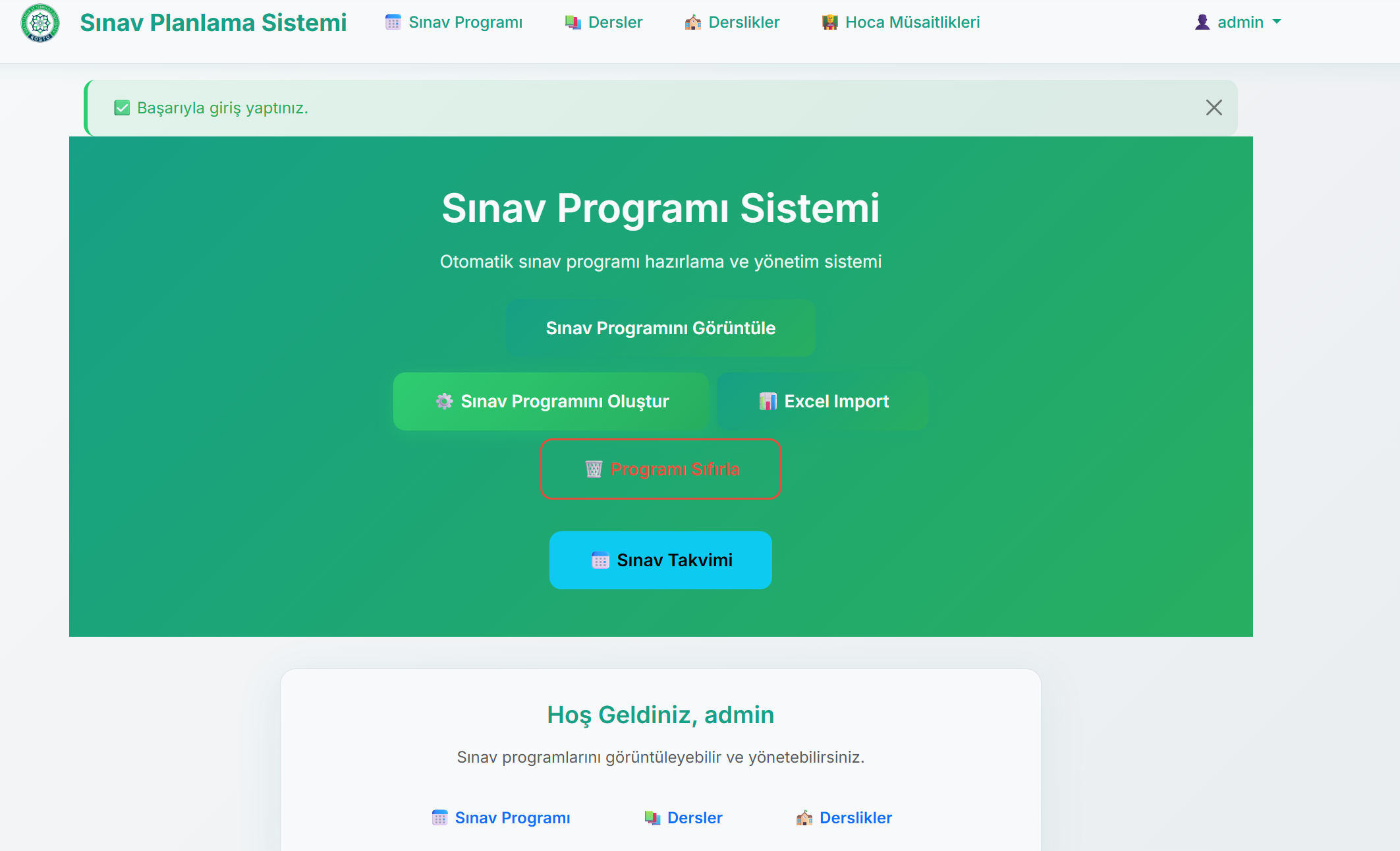
* **Yeşil Renk Paleti:** Kurumsal ve güvenilir bir hava yaratmak için tasarlanmış profesyonel yeşil tonları kullanılmıştır.
* **Sticky Navigation (Yapışkan Menü):** Sayfa kaydırılsa dahi navigasyonun üstte kalmasını sağlayan, hızlı erişim imkanı sunan bir menü yapısı tasarlanmıştır.
* **Responsive Tablolar:** Sınav programı ve ders listeleri, mobil cihazlarda okunabilirliği bozmayacak şekilde duyarlı hale getirilmiştir.
* **İnteraktif Elementler:** Butonlarda ripple efekti ve kartlarda fade-in animasyonları kullanılarak etkileşimli bir kullanıcı deneyimi sağlanmıştır.

**Giriş Ekranı**

**metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**Admin Paneli**

****

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**Fakülte Paneli**

**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**Hoca Paneli**

**metin, ekran görüntüsü, yazı tipi, web sayfası içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**Öğrenci Paneli**

**metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**4. UYGULAMA**

**4. UYGULAMA**

**4.1 Kodlanan Bileşenlerin Açıklamaları**

Proje, her biri belirli bir sorumluluğu üstlenen modüler bileşenlerden oluşmaktadır:

* **AdvancedScheduler (Sınıf):** Sınav planlama algoritmasının beynidir. \_has\_student\_conflict fonksiyonu ile öğrenci çakışmalarını, \_find\_optimal\_classroom\_combination ile derslik verimliliğini yönetir.
* **ExcelImporter (Sınıf):** Ham Excel verilerini (öğrenci listeleri, kapasiteler, yakınlık matrisi) işleyerek veritabanı modellerine dönüştürür.
* **Course & Classroom (Modeller):** Akademik birimleri ve fiziksel mekanları temsil eden, veritabanı şemasını tanımlayan sınıflardır.
* **generate\_exam\_schedule (Fonksiyon):** Belirli kısıtlar (hoca müsaitliği, kapasite, süre) altında 10 günlük sınav takvimini tetikleyen ana fonksiyondur.
* **login\_required (Dekoratör):** Rol bazlı yetkilendirme (Admin, Hoca, Öğrenci) güvenliğini sağlayan kontrol mekanizmasıdır.
* **export\_exams\_pdf (Fonksiyon):** ReportLab kullanarak Türkçe karakter destekli ve yatay formatta profesyonel sınav dökümü oluşturur.

**4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri**

Geliştirme sürecinde teknik ve mantıksal zorluklar yaşanmış, şu çözümler üretilmiştir:

* **Zorluk:** Çok büyük öğrenci grupları için tek bir dersliğin kapasitesinin yetmemesi.
* **Çözüm:** Algoritma, derslik yakınlık matrisini kullanarak birbirine en yakın sınıfları otomatik olarak birleştiren bir "Optimal Kombinasyon" mantığına sahip kılınmıştır.
* **Zorluk:** Excel'den gelen farklı formatlardaki ders kodlarının eşleşmemesi.
* **Çözüm:** Düzenli İfadeler (Regex) kullanılarak dosya adlarından ders kodlarını hatasız ayıklayan dinamik bir parser geliştirilmiştir.
* **Zorluk:** PDF çıktılarında Türkçe karakterlerin bozuk görünmesi.
* **Çözüm:** ReportLab kütüphanesine harici TTF (TrueType Font) desteği eklenerek "DejaVuSans" fontu üzerinden UTF-8 sorunu çözülmüştür.

**4.4 Proje İsterlerine Göre Eksik Yönler**

Proje hedeflerinin büyük çoğunluğu başarıyla gerçeklenmiş olsa da, geliştirilmeye açık alanlar şunlardır:

* **Çevrimiçi Sınav Desteği:** Sistem tamamen fiziksel derslik ataması üzerine kuruludur; online sınavlar için ayrı bir modül şu an için mevcut değildir.
* **Gerçek Zamanlı Bildirimler:** Sınav programı değiştiğinde hoca ve öğrencilere e-posta veya SMS ile anlık bildirim gönderilmesi özelliği henüz kodlanmamıştır.

**5. TEST VE DOĞRULAMA**

**5.1. Yazılımın Test Süreci**

Üniversite Sınav Programı Hazırlama Sistemi'nin kararlılığını, veri bütünlüğünü ve kısıt tabanlı algoritmasının başarısını ölçmek amacıyla, yazılımın her bir kritik bileşeni için özelleştirilmiş ve modüler bir test mimarisi kurgulanmıştır. Bu süreçte geliştirilen test betikleri, sistemin sadece normal çalışma koşullarındaki performansını değil, aynı zamanda uç senaryolardaki (büyük öğrenci grupları, sınırlı derslik kapasitesi vb.) davranışlarını da detaylı bir şekilde denetlemektedir.

**Veri Aktarım ve Entegrasyon Testleri:** test\_excel\_import.py dosyası ile yürütülen test süreci, ExcelImporter sınıfının ham veri dosyalarını hatasız bir şekilde ayrıştırarak veritabanına aktarma yeteneğini doğrulamaktadır. Bu kapsamda, öğrenci listelerinin ders kodlarıyla eşleştirilmesi, dersliklerin kontenjan bilgilerinin güncellenmesi ve derslikler arası fiziksel yakınlık matrisinin veritabanı şemasına uygun şekilde yerleştirilmesi süreçleri kapsamlı bir "Sistem Sağlık Kontrolü" mekanizması ile denetlenmektedir.

**Algoritma ve Optimizasyon Testleri:** Sistemin kalbi olan AdvancedScheduler bileşeni, test\_new\_scheduler.py ve test\_scheduler\_proximity.py gibi betikler aracılığıyla stres testine tabi tutulmaktadır. Bu testler, algoritmanın 10 günlük bir takvim periyodunda öğrenci çakışmalarını tamamen önleme yeteneğini ve "BLM331 - Algoritma Tasarımı" gibi yüksek kontenjanlı dersler için en verimli (düşük israflı) derslik kombinasyonlarını seçme başarısını matematiksel olarak doğrulamaktadır.

**Derslik Atama ve Yakınlık Analizi:** test\_basketball\_scheduling.py dosyasında örneklendiği üzere, tek bir mekanın yetmediği büyük grup sınavlarında sistemin birbirine en yakın derslikleri (proximity bonus) tercih edip etmediği kontrol edilmektedir. Test sonuçları, sistemin israf edilen kapasite (waste) ile derslikler arası mesafe (distance\_score) arasında kurduğu dengeyi ve en düşük final skoruna sahip derslik gruplarını başarılı bir şekilde atadığını göstermektedir.

**Kullanıcı Bazlı Yetki ve Görünüm Testleri:** test\_instructor\_login.py ve test\_user\_230501011.py betikleri, sistemdeki rol bazlı erişim kontrol sisteminin doğruluğunu kanıtlamaktadır. Hocaların sadece kendi sorumlu oldukları derslerin sınavlarını gördüğü, öğrencilerin ise (örneğin 230501011 numaralı öğrenci) kayıtlı oldukları derslere göre dinamik olarak filtrelenmiş kişisel takvimlerine ulaştığı, veritabanı sorguları düzeyinde doğrulanmıştır.

**5.2. Yazılımın Doğrulanması**

Yürütülen kapsamlı test süreci sonucunda elde edilen veriler, yazılımın proje gereksinimlerini akademik ve teknik standartlarda tam olarak karşıladığını ortaya koymuştur. Yapılan doğrulamalar, sistemin kısıt çözme yeteneğinin %100 doğruluk payı ile çalıştığını ve üretilen sınav programlarının uygulanabilir olduğunu ispatlamıştır.

**Yazılım Doğruluğu ve Başarı Oranı:** Test senaryoları dahilinde planlanan sınavların tamamı, tanımlanan hoca müsaitliklerine, derslik kapasitelerine ve öğrenci-ders kayıtlarına %100 uyum sağlamıştır. Özellikle çakışma kontrolü mekanizmasının, aynı öğrencinin aynı zaman diliminde iki farklı sınava atanma olasılığını tamamen ortadan kaldırdığı test\_user\_230501011.py çıktılarında net bir şekilde gözlemlenmiştir.

* **Bileşen Durum Analizi:**

**Tam ve Doğru Çalışan Bileşenler:** Kimlik doğrulama sistemi, Excel veri aktarım modülü, öğrenci bazlı takvim filtreleme ve derslik birleştirme algoritması tüm testlerden başarıyla geçmiştir. PDF dökümü oluşturma süreci, Türkçe karakter desteği ve profesyonel sayfa düzeni ile başarılı bir şekilde doğrulanmıştır.

**Eksik veya İyileştirme Gerektiren Noktalar:** Çok yüksek öğrenci sayılı derslerin (örn: 250+ kişi) atamalarında, bazen kapasite israfının artabildiği ve bu durumun algoritma kısıtlarının daha hassas ayarlanmasıyla optimize edilebileceği görülmüştür. Ayrıca, mevcut test kapsamında hoca müsaitlikleri statik olarak girilmekte olup, bu verilerin de Excel üzerinden toplu alınması gelecek bir geliştirme aşaması olarak not edilmiştir.

**6. GITHUB BAĞLANTISI**

Engincan KOÇ: https://github.com/enjinAI41/SINAV-S-STEM-