# Benim için Kaydet

### Git hub Adresi

https://github.com/sukruARSLAN2020/Benimicinkaydet

### Grubu oluşturan öğretmenlerin listesi

Onur DEMİRKAPI, Ayhan KOÇ, Erdal ÖZDOĞAN, İbrahim EDİZ

Şükrü ARSLAN, Mustafa UZAR, Musa BEKTES, Abdullah BOZAN, Resul BUTUNER, Rıdvan DEĞİRMENCİ, Fatma Funda Baysal KAYA, Ertuğrul ŞAHİN, İbrahim Can SARIÇİFTÇİ, Hasan YILMAZ

# Benim İçin Kaydet Analiz Raporu

### Kısa Özet

Proje pandemi döneminde öğrencilerin müfredat, ders, konu takiplerini yapmayı amaçlamaktadır. Ayrıca öğrenciler yaptıkları çalışmaları kaydedebilmekte ve gelişimlerini raporlayabilmektedirler.

### Problem Tanımı

Ne zaman yapmalıyım?

Öğrenciler sınıflarına ait ders konu quiz deneme sınavı gibi çalışmalarını takvime bağlamakta zamanı geldiğinde program onlara gerekli hatırlatmaları yapmaktadır. Bu işlemden sonra öğrenci yaptığı çalışmayı değerlendirmekte ve çalıştığı konuyu programa ekleyip daha sonra kontrol edebilmekte. Böylece konu yığılmaları önlemekte ve günü gününe öğrenci işlerini unutmadan yapabilmektedir. Ayrıca Eba dersleri, Zoom canlı derslerini de takvimine ekleyebilmektedir.

### Analiz Süreci

#### İhtiyaç Analizi

Öğrencilerin bazıları avantajlı yanlarında yüksek öğrenim görmüş velileri var. Bazıları ise bu imkandan yoksun. Fırsat ve imkan eşitliği acısından bu program onlara koçluk yapmayı hedeflemektedir. Öğrencilerin yapmak zorunda oldukları isleri onlara hatırlatacak

#### İçerik Analizi

Eba ders, konu ajandası,

Eba canlı ders ajandası,

Eba TV ajandası,

Zoom canlı ders ajandası,

Müfredat desteği ve içerik, takvim oluşturma ajandası,

Raporlama ve gelişim raporlama,

Konu özet veri tabanı,

Sınav ve sınav sonuçları veri tabanı

#### Durum Ortam Analizi

Veri tabanı

Grafik tasarım ara yüzü

Web desteği

Dosya işlemleri

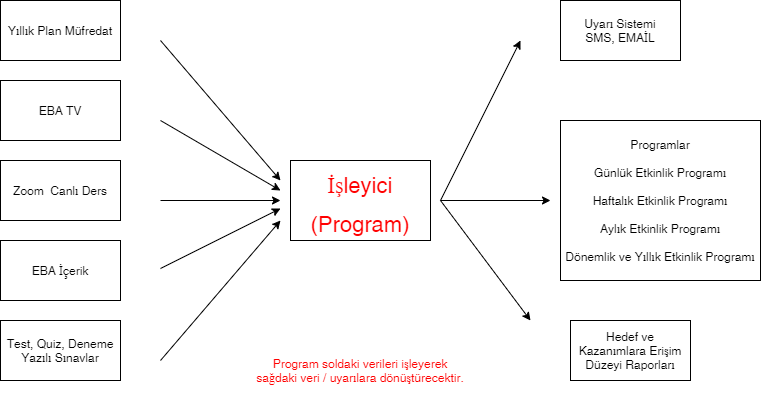
#### Kullanıcı Analizi

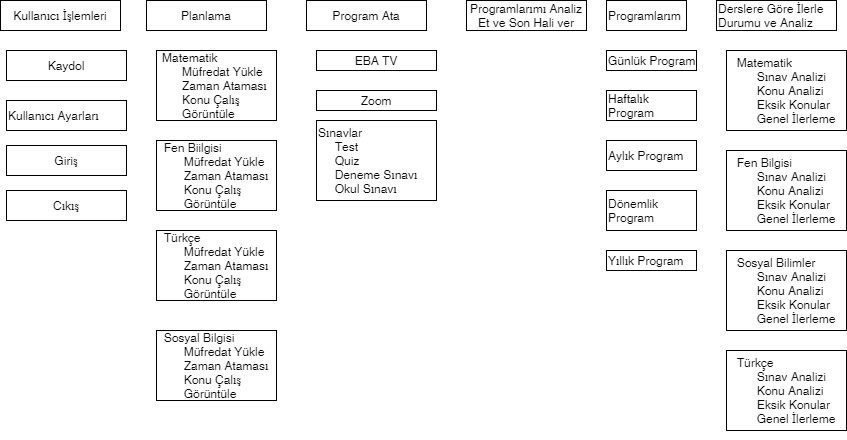
Yazılımı kullanacak olan eğitime gönül vermiş tüm programcılar

# Benim İçin Kaydet Tasarım Raporu

### Kısa Özet

Projenin Tasarım sürecinde uygulanan süreç adımlar:





Öncelikle sisteme kayıt olan proje grubu üyeleri ile iletişime geçildi. Projenin adımlarından herkesin haberdar edildi. Taslak üzerinde grup üyelerinin fikirleri alındı. İletişim kanalı kuruldu. Böylece gelişmelerden üyelerin haberdar olması sağlandı ve fikirleri alındı. Herkesin aldığı eğitim ölçüsünde gruba nasıl faydalı olabileceği soruldu ve iş bölümü yapıldı

Projeyle ilgili is bölümü:

1. Projenin taslak çalışması ve Raporlama çalışmaları: Şükrü ARSLAN, Mustafa UZER, Rıdvan DEGIRMENCI

2. Veri Tabanı Analiz çalışması ve veri tabanı oluşturulması : Resul BUTUNER

3. Grafik Tasarım ve Ara yüz: Ertuğrul SAHIN, Şükrü ARSLAN, Fatma Funda Baysal KAYA

4. Zaman Çizelgesi: Şükrü ARSLAN

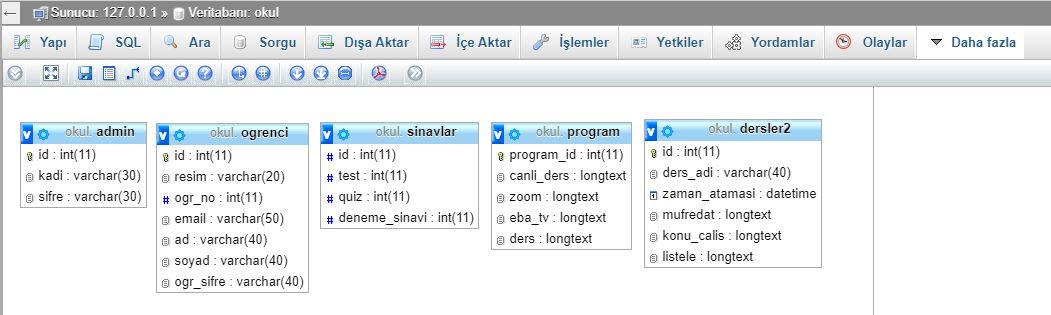
5. Kod Analizi ve Projenin yazım aşamasına getirilmesi : Musa BEKTES

***6-7 Aralık tarihleri arasında grup üyeleri ile tekrar istişare edilerek program modüllere ayrılacak ve grup üyelerinin yeni görevleri netleştirilecek.***

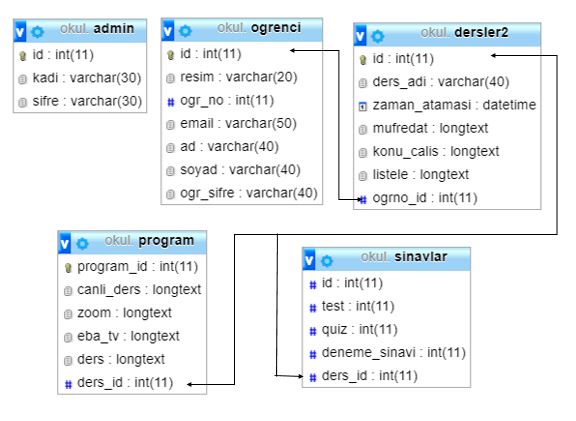
## Veri Tasarımı (Resul BÜTÜNER)

Projede veri tabanı kullanıldığı durumda veri tabanı diyagramı :

Projede örnek olarak dört adet ham veri dosyası bulunmaktadır. Matematik, Fen Bilgisi, Sosyal Bilimleri Türkçe bunlar sistem tarafından işlenerek programın kullanabileceği veri tabanlarına dönüştürülecektir. Ayrıca kullanıcı EBA ders Saatlerini, EBA Canlı ders saatlerini, Zoom ders saatlerini sisteme girecek ve program veri tabanlarına işleyecektir. Kullanıcı zamanı geldiğinde işlediği konuyla ilgili kendi notlarının sisteme işleyecektir. Kullanıcı ayrıca test, quiz, deneme sınavı tarihleri belirleyip zamanı geldiğinde aldığı puanı sisteme işleyecektir. Süreç içinde de genel ilerleme durumunu raporlayabilecektir.



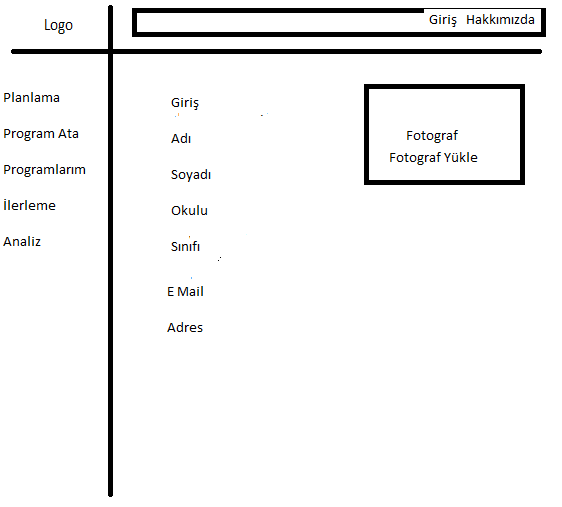
İlişkisel veri tabanı tanımı :



## Ara yüz Tasarımı (Ertuğrul SAHIN, Şükrü ARSLAN, Fatma Funda Baysal KAYA)

Kullanıcı ara yüzüne ait tasarımları:

1. Kullanıcı Girişi ve Kayıt



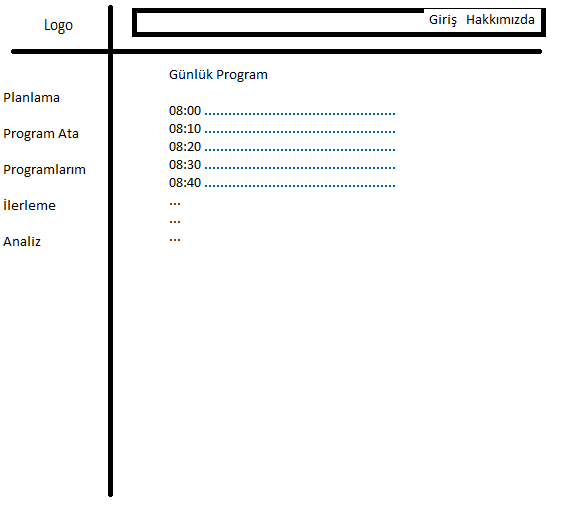
1. Aktivite Tanımlama



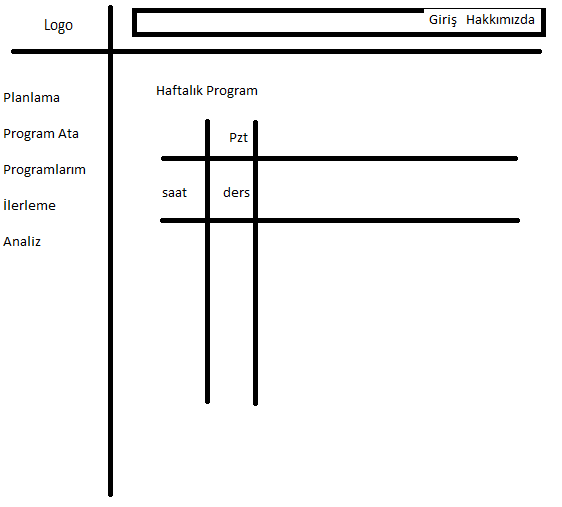
1. Kayıtlı Aktiviteler / Görüntüle :



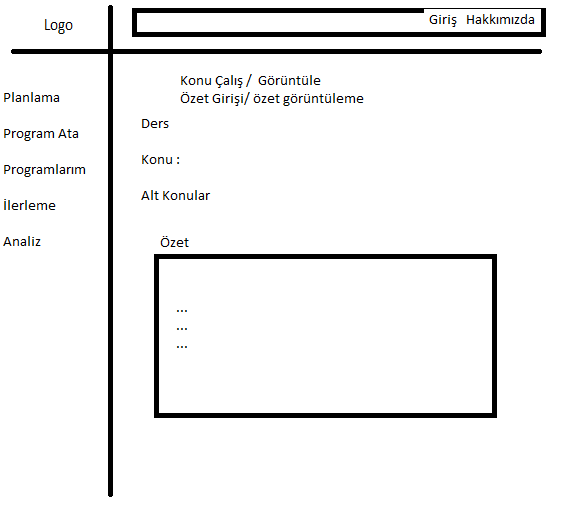
1. Programlarım: (Günlük Program)



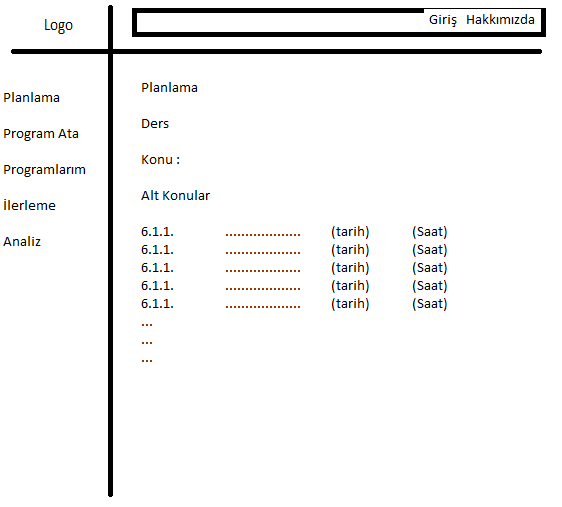
1. Programlarım (Haftalık Program)



1. Konu Çalış

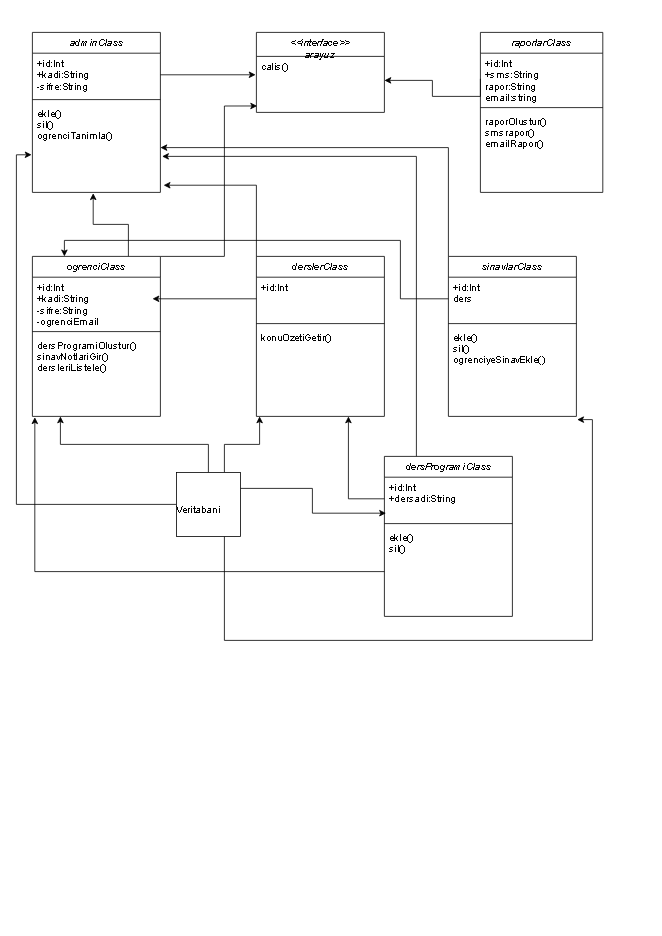


1. Müfredat dağıtma :



## Kod Tasarımı (Musa BEKTES)

Yazılım geliştirme süreci sırasında Nesne Yönelimli Programlama dikkate alınarak (Class Diyagram) tasarımlar:



## Zaman Çizelgesi

Projenin rapor yazım süreçleri de dahil edilerek gerektiğinde iç içe geçmiş süreçlerinde ifade edilebileceği bölümdür. (Gannt Chart)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ZAMAN ÇİZELGESİ |  | 10.11.2020 | 11.11.2020 | 12.11.2020 | 13.11.2020 | 14.11.2020 | 15.11.2020 | 16.11.2020 | 17.11.2020 | 18.11.2020 | 19.11.2020 | 20.11.2020 | 21.11.2020 | 22.11.2020 | 23.11.2020 | 24.11.2020 | 25.11.2020 | 26.11.2020 | 27.11.2020 | 28.11.2020 | 29.11.2020 | 30.11.2020 | 01.12.2020 | 02.12.2020 | 03.12.2020 | 04.12.2020 | 05.12.2020 | 06.12.2020 | 07.12.2020 | 08.12.2020 | 09.12.2020 | 10.12.2020 | 11.12.2020 | 12.12.2020 | 13.12.2020 | 14.12.2020 | 15.12.2020 | 16.12.2020 | 17.12.2020 | 18.12.2020 | 19.12.2020 | 20.12.2020 | 21.12.2020 | 22.12.2020 | 23.12.2020 | 24.12.2020 | 25.12.2020 | 26.12.2020 | 27.12.2020 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Grupların Netleştirilmesi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analiz Raporu |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Kısa Özet |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Problemin Tanımı |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analiz Süreci |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | İhtiyaç Analizi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | İçerik Analizi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Durum Ortam Analizi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Kullanıcı Analizi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Tasarım Süreci |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Kısa Özet |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Veri Tasarımı |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ara Yüz Tasarımı, |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Kod Tasarımı |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Zaman Çizelgesi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Gerçekleştirme |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Karşılaşılan Sorunlar ve Uygulanan Çözümler |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Proje Bileşenleri ve Görevleri |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Github Yükleme Süreci |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test Süreci |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Karşılaşılan Sorunlar ve Uygulanan Çözümler |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Kullanılan Mödüller |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Benim için Kaydet Gerçekleştirme Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

Projenin gerçekleştirme sürecine karşılaşılan sorunlar ve gidermek için uygulanan çözümlerin ifade edildiği bölümler.

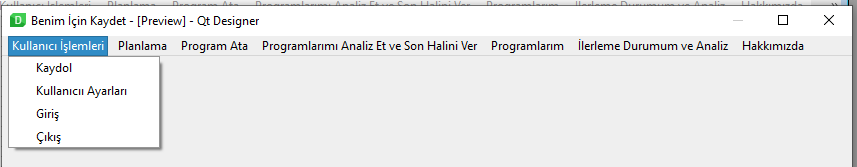
## Proje Bileşenleri ve Görevleri:

Programa ait dokümantasyonunun taslak halinde ortaya çıktığı bölümdür.

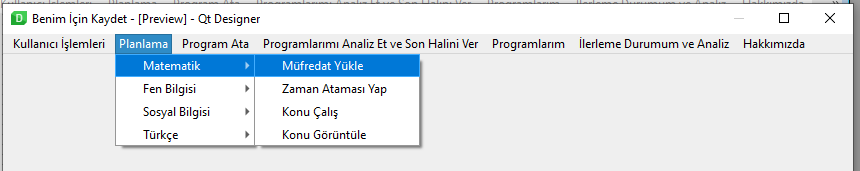
1. Ham veri Text Dosyaları :
2. 6. Sınıf Matematik ham veri dosyası
3. 6. Sınıf Fen Bilgisi ham veri dosyası
4. 6. Sınıf Türkçe ham veri dosyası
5. 6. Sınıf Sosyal Bilgiler ham veri dosyası

Ham veri dosyaları işlendikten sonra programın kullanabileceği BenimİçinKaydet (BIK.db) dosyasına dönüştürülür.

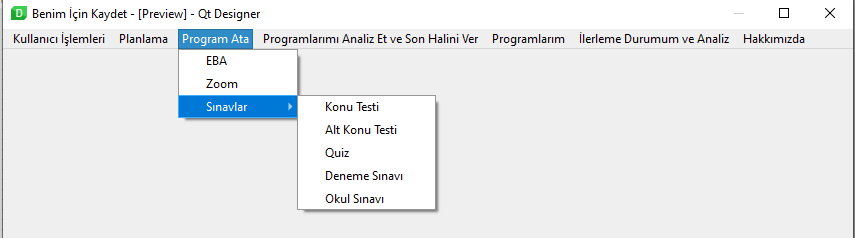
1. BIK içindeki tablolar ve görevleri
2. Fen Bilgisi, Matematik, Sosyal Bilgiler, Türkçe dersleri: Fen Bilgisi, Matematik, Sosyal Bilgiler, Türkçe dersleri ne ait tablolarda konu, alt konu, tarih, saat, sure, ders işlenme durumu, uygulamanın hangi platformda tutulduğu, ve bu konulara ait sınavların ıd lerinin tutulduğu veri tabanı kısımları (ayrı ayrı 4 tablo)
3. Yönetici(tablo): programın temel ayarlarını yapabilecek olan yöneticilerin bilgilerinin tutulduğu veri tabanı tablosu,
4. Öğrenci(tablo) : öğrencilerle ilgili temel bilgilerin tutulduğu veri tabanı kısmı.
5. Sınavlar (tablo) : Öğrencilerin konu, alt konulardan kendisine uyguladığı sınavların sonuçlarını tutan veri tabanı tablosu,
6. Platform(tablo) : Öğrencilerin EBA, zoom vb çeşitli platformlarda kendisine uyguladığı aktivitelerin verilerinin tutulduğu veri tabanı tablosu
7. Ara yüz tanımlamaları ve görevleri :



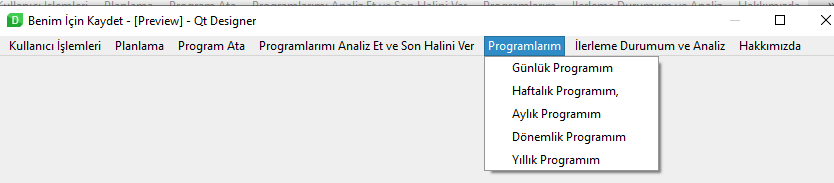
1. Kullanıcı işlemleri :
2. Kaydol : öğrencinin sisteme kayıt olmasını,
3. Kullanıcı ayarları : öğrencinin sistemin kullanacağı verileri kaydedebileceği veya güncelleyebileceği kısım,
4. Kayıtlı öğrencilerin sisteme giriş yapabilmesini,
5. Programdan çıkış bölümü



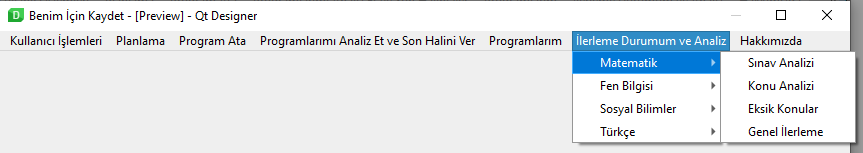
1. Planlama : öğrencilerin derslere ait müfredat yüklemesi, zaman ataması yapması, Konu çalışması ve çalıştığı konuyu text formatında sisteme yükleyebildiği, konu görüntüle kısmında ise öğrencinin çalıştığı konuya ait verileri ekranda gözden geçirebilmesini sağlayan kısımlarda oluşan bölüm,



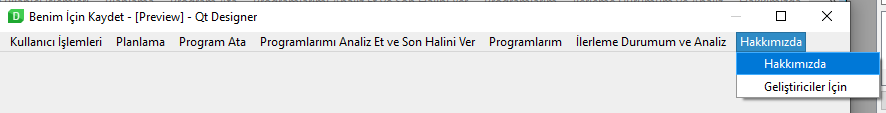
1. Program Ata : öğrencilerin programa EBA, Zoom, Sınav tarihi ie ilgili aktivitelerini planlayabileceği kısım
2. Programı Analiz ve son Halini ver komutunda ise programın genel olarak veri tabanı dosyalarının gözden geçirildiği ve sistem dosyalarında aksaklık olup olmadığının kontrol edildiği kısım



1. Sisteme kayıtlı verilere göre günlük, haftalık aylık programların kullanıcıya gösterildiği bölüm



1. Kullanıcının derslerde ne kadar ilerlediğinin gösterildiği kısım. Burada sınavlarla, konularla, eksik konularla ve genel ilerleme durumu verilerine ulaşılabilecek.



1. Bu kısımda programı hazırlayan grup bilgileri ve GitHuba yüklendikten sonra geliştiriciler için dokümantasyon kısmının bulunduğu kısım.
2. Programın grup üyelerince daha rahat yapılabilmesi için modüllere ayrıldı. Bu modüller ve görevleri
   1. Ana program : alt modüllerden gelen bilgileri derleyip ara yüzle entegrasyonu,
   2. Dersler : derslerle ile ilgili işleri,
   3. Öğrenci : öğrenci ile ilgili işleri,
   4. Platformlar : platformlarla ilgili işleri,
   5. Sınavlar : sınavlarla ilgili işleri,
   6. Veri tabanı : veri tabanı ile ilgili iş ve işlemleri,
   7. Yönetici : (sistem admin) sistem yöneticisinin yapacağı işleri

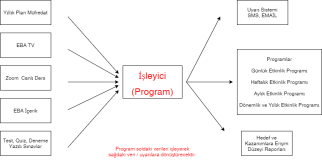
Kapsamaktadır.

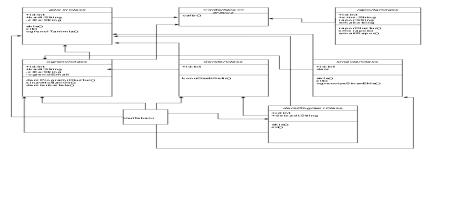
Bu işlerle ile ilgili iş bölümü :

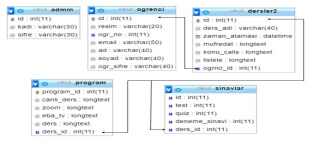
1. Ana program : Mustafa UZAR,
2. Yönetici :Resul BUTUNER,
3. Veri tabanı : şükrü ARSLAN,
4. Öğrenciler : Rıdvan DEĞİRMENCİ,
5. Dersler : Ertuğrul ŞAHİN,
6. Platformlar : Fatma Funda Baysal KAYA
7. Sınavlar : İbrahim SARIÇİFTÇİ
8. Raporlar : Abdullah BOZAN
9. Arayüz Tasarımı : Musa BEKTEŞ

## Github Yükleme Süreci

Raporlama aşamasında elde edilen verilerden uygun olanlarından ayrıca proje tasarım menümüzde geliştiriciler için sekmesi içerisine veriler elden geldiğince konacaktır.







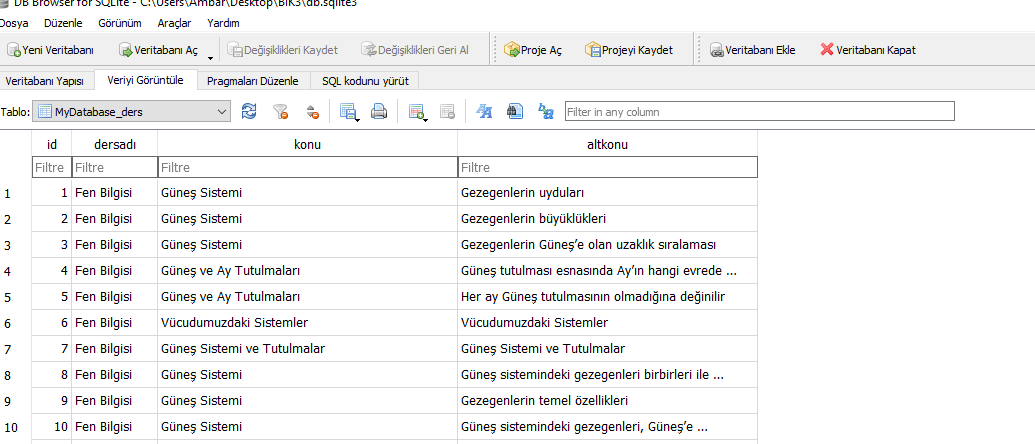
# Benim için Kaydet Test Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

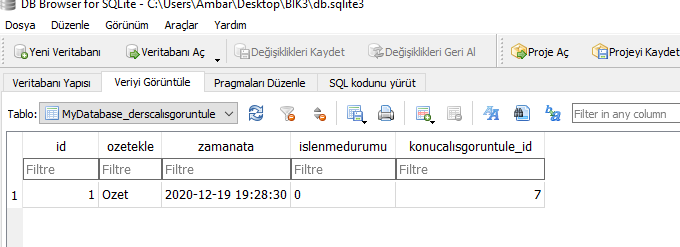
Yazılım projesinin çalıştırılması ve test edilmesi süresince karşılaşılan sorunlar ve uygulanan çözümler yer almaktadır.

Proje gerçekleştirme esnasında python ile ilgili eğitim çalışmalar devam etmekteydi. Projenin alt yapısına bakıldığında projenin django alt yapısına daha uygun olduğu fark edildi. Grup üyeleriyle birlikte programın djangoya taşınmasına ve uygulamanın burada yapılmasına kara verildi. Özellikle djangonun hazır admin ve yönetici paneli bu kararın alınmasında etkili oldu. Bu arada veri tabanı yapısı tekrar gözden geçirildi.

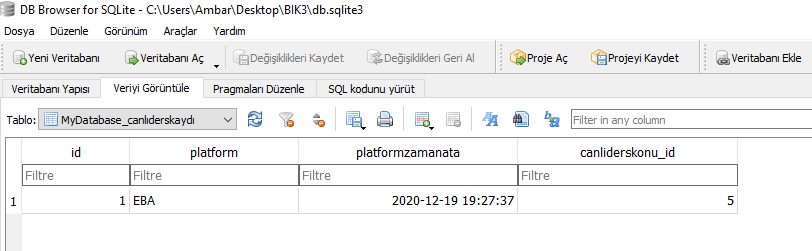
Ders tablosu veri yapısı :



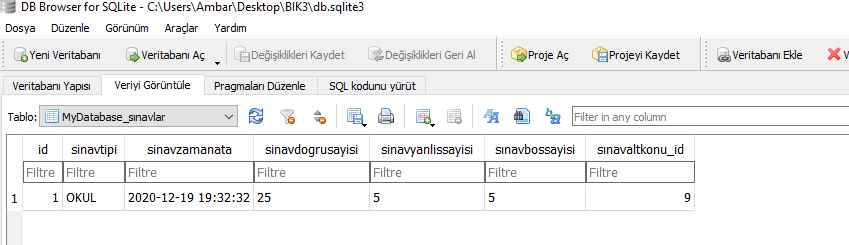
Ders çalış görüntüle tablosu veri yapısı :



Canlı ders kaydı tablosu veri yapısı :



Sınavlar tablosu veri yapısı :



Veri tabanında ders tablosu Foreignkey((yabancı anahtar olarak kullanıldı.)

from django.db import models

class Ders(models.Model):

    dersadı=models.CharField(max\_length=15)

    konu=models.CharField(max\_length=50)

    altkonu=models.CharField(max\_length=200)

    def \_\_str\_\_(self):

        return self.dersadı

class DersCalısGoruntule(models.Model):

    konucalısgoruntule=models.ForeignKey('Ders',on\_delete=models.CASCADE)

    ozetekle=models.TextField()

    zamanata=models.DateTimeField()

    islenmedurumu=models.BooleanField(default='false')

    #def \_\_str\_\_(self):

    #   return self.konucalısgoruntule

class CanlıDersKaydı(models.Model):

    canliderskonu=models.ForeignKey('Ders',on\_delete=models.CASCADE)

    platform=models.CharField(max\_length=15)

    platformzamanata=models.DateTimeField()

    def \_\_str\_\_(self):

        return self.platform

class Sınavlar(models.Model):

    sınavaltkonu=models.ForeignKey('Ders',on\_delete=models.CASCADE)

    sinavtipi=models.CharField(max\_length=25)

    sinavzamanata=models.DateTimeField()

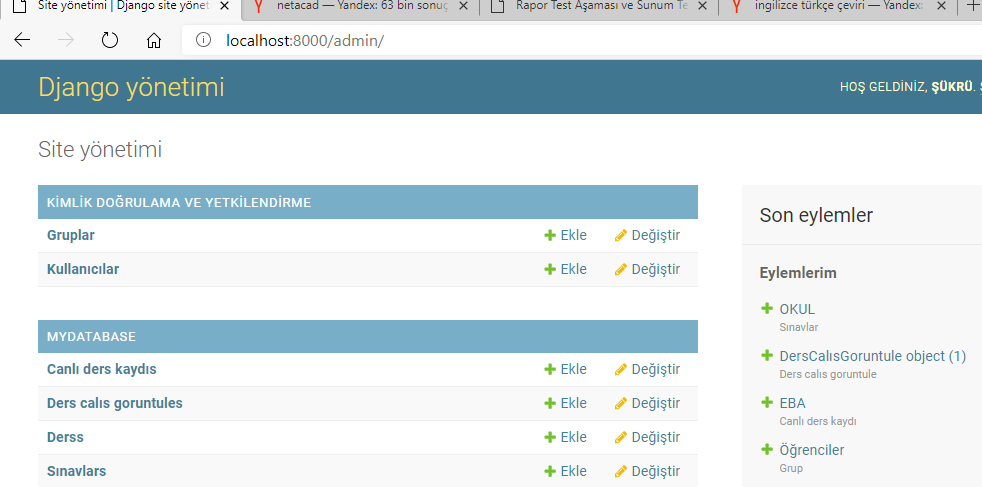
    sinavdogrusayisi=models.PositiveSmallIntegerField()

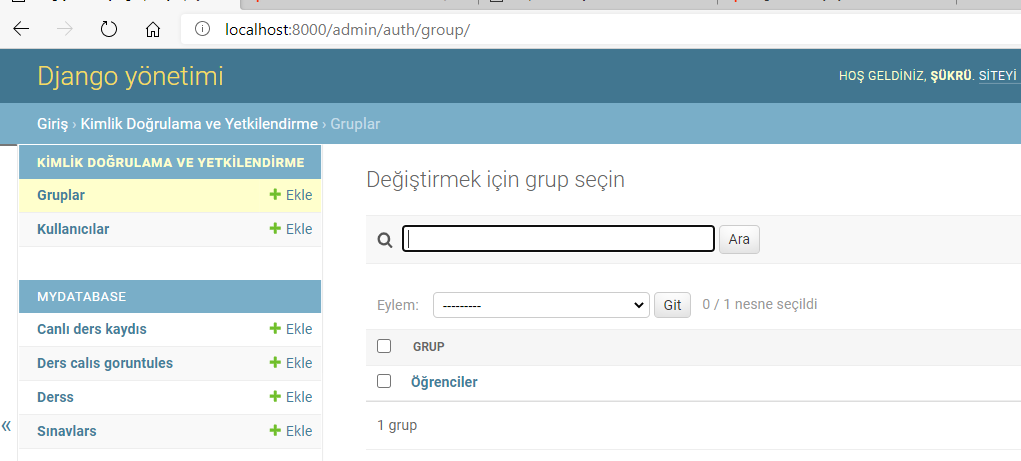
    sinavyanlissayisi=models.PositiveSmallIntegerField()

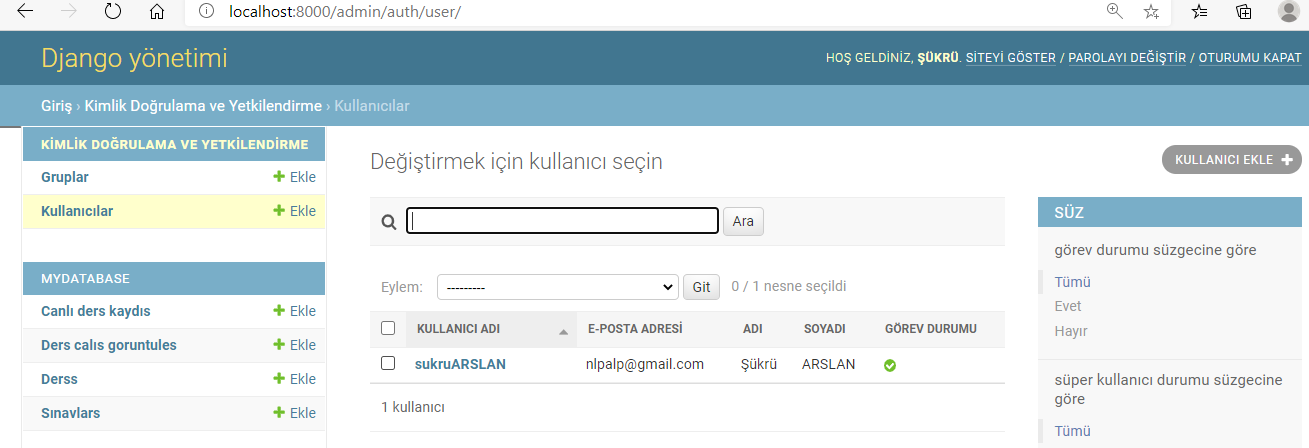
    sınavbossayisi=models.PositiveSmallIntegerField()

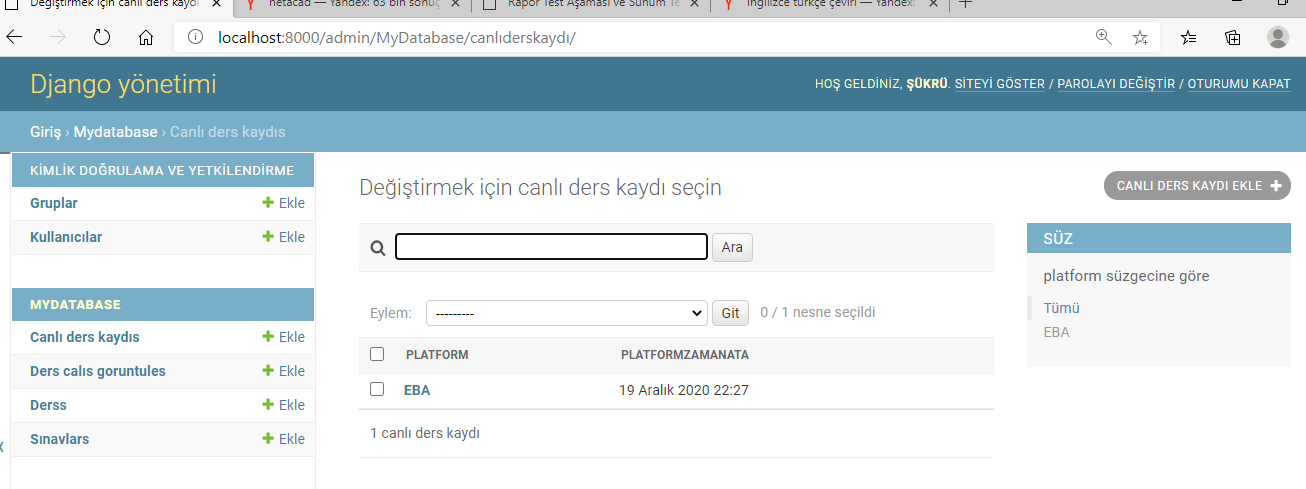
    def \_\_str\_\_(self):

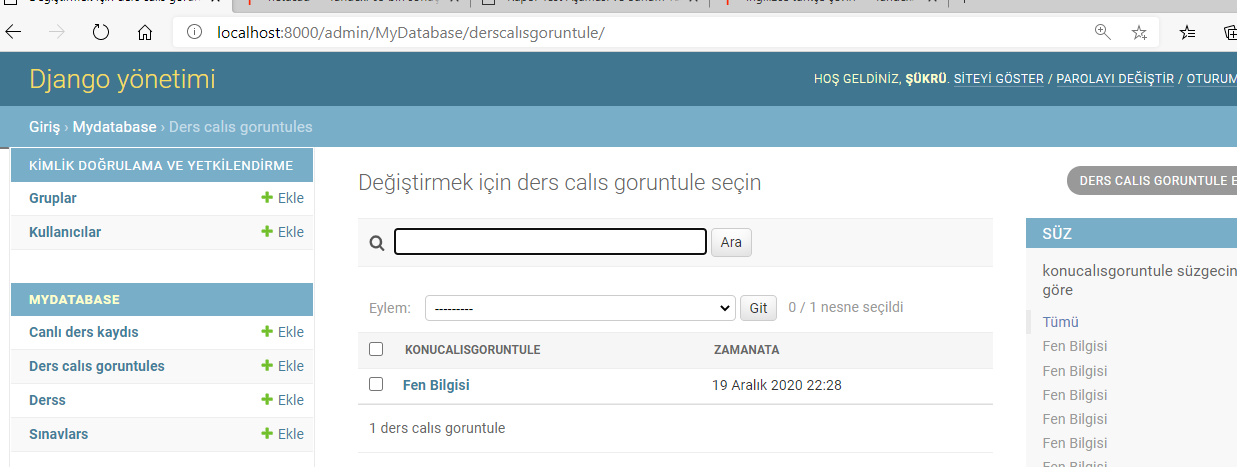
        return self.sinavtipi













Yukarıdaki admin paneliyle ilgili filtreler ve sistem ayarlarının bulunduğu py modülü

from django.contrib import admin

from .models import Ders,DersCalısGoruntule,CanlıDersKaydı,Sınavlar

class PostDers(admin.ModelAdmin):

    list\_display = ['dersadı', 'konu','altkonu']

    list\_display\_links = ['altkonu']

    list\_filter = ['dersadı']

    search\_fields = ['dersadı', 'konu']

    #list\_editable = ['altkonu']

    class Meta:

        model = Ders

admin.site.register(Ders, PostDers)

class PostDersCalısGoruntule(admin.ModelAdmin):

    list\_display = ['konucalısgoruntule', 'zamanata']

    list\_display\_links = ['konucalısgoruntule']

    list\_filter = ['konucalısgoruntule']

    search\_fields = ['konucalısgoruntule']

    #list\_editable = ['zamanata']

    class Meta:

        model = DersCalısGoruntule

admin.site.register(DersCalısGoruntule,PostDersCalısGoruntule)

class PostCanlıDersKaydı(admin.ModelAdmin):

    list\_display = ['platform','platformzamanata']

    list\_display\_links = ['platform']

    list\_filter = ['platform']

    search\_fields = ['canlıderskonu', 'platform']

    #list\_editable = ['zamanata']

    class Meta:

        model = CanlıDersKaydı

admin.site.register(CanlıDersKaydı,PostCanlıDersKaydı)

class PostSınavlar(admin.ModelAdmin):

    list\_display = ['sınavaltkonu','sinavtipi']

    list\_display\_links = ['sinavtipi']

    list\_filter = ['sinavtipi']

    search\_fields = ['sınavaltkonu', 'sinavtipi']

    #list\_editable = ['zamanata']

    class Meta:

        model = Sınavlar

admin.site.register(Sınavlar,PostSınavlar)

Programda yukarıda anlatılan sistemler doğru çalışmaktadır. Ancak grup üyeleri olarak bizler süre sınırlı olduğu gerekli kodların bir kısmı hazırlanmış fakat template mantığı için zamana ihtiyaç olduğundan admin panelinden web ara yüzüne aktarım tamamlanamamıştır.

from django.shortcuts import render, HttpResponse, get\_object\_or\_404, HttpResponseRedirect, redirect, Http404

from .models import MyDatabase

from .forms import PostForm, CommentForm

from django.contrib import messages

from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger

from django.db.models import Q

def MyDatabase\_index(request):

    MyDatabase\_list = MyDatabase.objects.all()

    query = request.GET.get('q')

    if query:

        post\_list = post\_list.filter(

            Q(title\_\_icontains=query) |

            Q(content\_\_icontains=query) |

            Q(user\_\_first\_name\_\_icontains=query) |

            Q(user\_\_last\_name\_\_icontains=query)

        ).distinct()

    paginator = Paginator(post\_list, 5)  # Show 5 contacts per page

    page = request.GET.get('page')

    try:

        posts = paginator.page(page)

    except PageNotAnInteger:

        # If page is not an integer, deliver first page.

        posts = paginator.page(1)

    except EmptyPage:

        # If page is out of range (e.g. 9999), deliver last page of results.

        posts = paginator.page(paginator.num\_pages)

    return render(request, "MyDatabase/index.html", {'posts': posts})

def MyDatabase\_detail(request, slug):

    post = get\_object\_or\_404(Post, slug=slug)

    form = CommentForm(request.POST or None)

    if form.is\_valid():

        comment = form.save(commit=False)

        comment.post = post

        comment.save()

        return HttpResponseRedirect(post.get\_absolute\_url())

    context = {

        'post': MyDatabase,

        'form': form

    }

    return render(request, "MyDatabase/detail.html", context)

def post\_create(request):

    if not request.user.is\_authenticated():

        # Eğer kullanıcı giriş yapmamış ise hata sayfası gönder

        return Http404()

    form = PostForm(request.POST or None, request.FILES or None)

    if form.is\_valid():

        post = form.save(commit=False)

        post.user = request.user

        post.save()

        messages.success(request, "Başarılı bir şekilde oluşturdunuz.", extra\_tags='mesaj-basarili')

        return HttpResponseRedirect(post.get\_absolute\_url())

    context = {

        'form': form

    }

    return render(request, "MyDatabase/form.html", context)

def MyDatabase\_update(request, slug):

    if not request.user.is\_authenticated():

        # Eğer kullanıcı giriş yapmamış ise hata sayfası gönder

        return Http404()

    post = get\_object\_or\_404(Post, slug=slug)

    form = PostForm(request.POST or None, request.FILES or None, instance=post)

    if form.is\_valid():

        form.save()

        messages.success(request, "Başarılı bir şekilde güncellediniz.")

        return HttpResponseRedirect(post.get\_absolute\_url())

    context = {

        'form': form

    }

    return render(request, "MyDatabase/form.html", context)

def MyDatabase\_delete(request, slug):

    if not request.user.is\_authenticated():

        # Eğer kullanıcı giriş yapmamış ise hata sayfası gönder

        return Http404()

    post = get\_object\_or\_404(Post, slug=slug)

    post.delete()

    return redirect("MyDatabase:index")

## Test Sürecinde Kullanılan Modüller (Varsa)

Proje test sürecinde gerektiğinde farklı modüller kullanılarak test çalışması gerçekleştirilmektedir. Proje sürecinde eğer bu modüllerden herhangi birini kullandıysanız. Modülü kullanırken yaptığınız kodlama bu bölümde yer almaktadır.

## Değerlendirme Kriterleri

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KRİTERLER** | Ekip Üyeleri | Eğitmen 1 | Eğitmen 2 | Eğitmen 3 | Ortalama |
| 1. Analiz Raporunun Tamamlanması | Şükrü ARSLAN, Mustafa UZER |  |  |  |  |
| 1. Tasarım Raporunun Tamamlanması | Şükrü ARSLAN, Musa BEKTES,  Resul BUTUNER |  |  |  |  |
| 1. Gerçekleştirim Raporunun Tamamlanması | Şükrü ARSLAN, Rıdvan DEĞİRMENCİ |  |  |  |  |
| 1. Gantt Diyagramı | Musa BEKTES |  |  |  |  |
| 1. Ara yüz tasarımı | Tüm Grup Üyeleri |  |  |  |  |
| 1. Veri Tasarımı-Sınıf Tasarımı | Şükrü ARSLAN, Musa BEKTES,Resul BUTUNER |  |  |  |  |
| 1. Kullanıcı Yardım Dokümanı | Tüm Grup Üyeleri |  |  |  |  |
| 1. Programın Çalıştırılması | Tüm Grup Üyeleri |  |  |  |  |
| 1. Yazılım Test Çalışması | Tüm Grup Üyeleri |  |  |  |  |