



T.C
SİVAS CUMHURİYET ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



STAJ DEFTERİ

2020141039
Samet Ertuğrul Büyükevli

ÖĞRETİM YILI
2023/2024

**T.C.
CUMHURİYET ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
STAJ YÖNERGESİ**

Amaç MADDE 1

Bu yönernenin amacı; Cumhuriyet Üniversitesi Mühendislik Fakültesi öğrencilerinin öğrenim sürecinde kazandıkları teorik ve uygulamalı bilgilerin değerlendirilmesi ve pekiştirilmesi amacıyla eğitim-öğretimin zorunlu bir parçası olan stajlar ile ilgili usul ve esasları belirlemektir.

Kapsam MADDE 2

Bu yönerge Cumhuriyet Üniversitesi Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği gereği, Mühendislik Fakültesi öğrencilerinin öğrenim süresince yapmakla yükümlü oldukları staj çalışmalarının ilkelerini, uygulama ve değerlendirme kurallarını kapsar.

Dayanak MADDE 3

Bu yönerge 2547 sayılı Yükseköğretim Kanunu ve Cumhuriyet Üniversitesi Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği hükümlerine göre hazırlanmıştır.

Staj Süresi ve İçeriği MADDE 4

- (1) Mühendislik Fakültesi öğrencilerinin zorunlu staj süresi en az 45 iş günüdür.
- (2) Öğrenciler, stajlarını 2. yarıyılidan itibaren bir seferde en az 10, en fazla 30 iş günü olarak yapabilirler. Hangi dönemde ne kadar staj yapılacağına bölüm staj komisyonu karar verir.
- (3) Zamanında yapılamayan ve/veya reddedilen stajlar, eğitim-öğretim dönemleri, yaz okulu ve yarıyıl sonu sınav haftaları dışında kalan sürelerde, daha üst sınıflarda veya ders alma zorunluluğunun olmadığı eğitim-öğretim dönemlerinde yapılabilir.
- (4) Bir haftadaki staj iş günü sayısı 5 gündür. Ancak staj yapılan kurumda cumartesi günleri de çalışılıyorsa belgelendirilmesi şartıyla staj gün sayısı 6 gün olarak değerlendirilir. Kurumda pazar günleri çalışılsa bile, pazar günleri ve resmi tatil günleri staj iş gününden sayılmaz.
- (5) Staja fiilen ve kesintisiz devam zorunluluğu vardır. Mazereti nedeni ile de olsa devam edilemeyen staj günleri, toplam staj gününden düşülür.
- (6) Her bölüm, kendi staj içeriğini belirler ve öğrencilerine ilan eder.

Bölüm Staj Komisyonu ve Görevleri

MADDE 5 –

- (1) Bölüm Staj Komisyonu, en az 3 asil öğretim üyesi ve/veya görevlisinden oluşur ve her yıl Bölüm Akademik Kurulu tarafından seçilir.
- (2) Bölüm Staj Komisyonu, stajlarla ilgili programları ve esasları hazırlar, staj yerlerinin uygunluğunu belirler, staj yerlerini dağıtır, staj dosyası ve staj sicil fışlarını inceleyerek veya inceleterek yapılan stajları değerlendirir ve karara bağlar.

Staj Yeri ve Dağıtımı MADDE 6

- (1) Staj yapılacak yerlerin öğrenciye uygulamalı çalışma alanında bilgi kazandıracak yeterlikte ve organizasyon düzeyinde olması gereklidir. Bölümün sağladığı ya da öğrencilerin bulduğu staj yerlerinin uygunluğunu belirlenmesinde Staj Komisyonu tam yetkilidir.
- (2) Staj, bölüm staj komisyonunun uygunluğunu kabul ettiği alan ile ilgili yurt içi veya yurt dışı resmi veya özel kurumlarda yapılır.
- (3) Staj yapılacak kurumda, alanında/alanına yakın en az 1 mühendis bulunmalıdır.
- (4) Öğrenciler, Bölüm Başkanlığının teklifi ve Dekanlığın onayı ile stajlarının en fazla 20 iş gününü Mühendislik Fakültesinde yapabilirler.
- (5) Öğrenci kendi mühendislik alanı ile ilgili bir iş yerinde çalışıyor ise, bölüm staj komisyonunun uygun görmesi halinde çalıştığı kurumda staj yapabilir.
- (6) Bölümce sağlanan staj yerleri Bölüm Başkanlığı tarafından duyurulur. Öğrenciler duyurulan staj yerlerinden üçünü istek sırasına göre belirler ve bir dilekçe ile Bölüm Başkanlığına başvurur. Staj yerlerinin dağıtımı Staj Komisyonunda öğrencilerin istek sırası ve başarı durumları da göz önüne alınarak yapılır ve duyurulur.
- (7) Staj yerlerini kendileri bulan öğrenciler, staj yapmak istedikleri kuruluştan ilgili Bölüm Başkanlığına hitaben yazılmış staja kabul edildiğini, staj konusunu ve süresini belirten bir yazı getirir.

Staj Muafiyeti MADDE 7

- (1) Fakülteye ilk kez kayıt yaptıran bir öğrenci, kayıt olduğu ilk yarıyıl/yılın ikinci haftasının sonuna kadar bölümne başvurarak, daha önce okumuş olduğu bir yükseköğretim kurumunda alanı ile ilgili staj yapmış ise muafiyet isteğinde bulunabilir.
- (2) Staj muafiyeti için başvuruda bulunan öğrenci, daha önce yapmış olduğu stajını resmi evrak ile belgelemek zorundadır.

(3) Bölümler, öğrencinin muafiyet başvurusunu, başvuru bitim tarihinden itibaren bir hafta içinde bölüm staj komisyonuna inceletir, Bölüm Kurulu olarak kararını verir ve Fakülte Yönetim Kurulu'na sunar.

Staj Defteri/Dosyası, Gizli Sicil ve Staja Başlama

MADDE 8

- (1) Öğrenci stajla ilgili işlemlerini, staja başlamadan önceki yarıyılın sonuna kadar tamamlayarak Bölüm Başkanlığına başvurur.
- (2) Staj yerleri belirlenen öğrenciler ilgili Bölüm Başkanlığından staj yapılacak kuruluşa hitaben yazılan bir yazı ile fotoğraflı "Staj Defteri" ve "Gizli Sicil" belgesini alır, Staj Defteri ile Gizli Sicil belgesini Dekanlıkta onaylatır.
- (3) Her öğrenci staja başlamadan önce yasal süresi içinde sigorta işlemlerini tamamlatır.
- (4) Öğrenci staj kabul yazısında ve sigorta başlangıcında belirtilen tarihte staj yapacağı kuruluşa giderek gerekli belgeleri işyeri amirine sunar ve staja başlar.

Staj Sırasındaki ve Bitimindeki İşlemler

MADDE 9

- (1) Staj defterinde staj yapılan kuruma ayrılan yer ile öğrencinin "Staj Programı" Kurum Amirince onaylanır.
- (2) Öğrenci staj defterindeki "Günlük Çalışma Programı" bölümüne, günlük yaptığı işleri yazarak staj yaptığı kurumun alanında/alanına yakın mühendise onaylatır.
- (3) Öğrenci, her staj dönemi için ayrı bir staj raporu hazırlar. Bu işlem için bir veya daha fazla staj dosyası kullanılabilir.
- (4) Staj yeri amiri; öğrencinin devamı, ilgisi, başarısı ve diğer durumlarına ilişkin bilgileri gizli sicil fişine işleyip imzalandıktan sonra, kapalı zarf içerisinde öğrenci ile veya taahhütlü olarak ilgili bölüm başkanlığına gönderir.
- (5) Staja başlayan bir öğrenci, staj komisyonu, bölüm başkanlığı ve dekanlığın izni olmadan staj yerini değiştiremez.
- (6) Öğrenci, zamanında başlanılmayan/tamamlanmayan stajların sigorta çıkış işlemleri için Fakülte Staj Birimine bilgi vermek zorundadır.
- (7) Tüm işlemleri tamamlandığı halde mazeretsiz olarak stajına gitmeyen ve Fakülte Staj Birimine bildirmeyen öğrencilere o dönem için yeniden staj işlemi yapılmaz.

Staj Dosyasının Teslimi ve Değerlendirilmesi

MADDE 10

- (1) Öğrenciler, yaptıkları stajlar için hazırladıkları staj dosyasını ve staj sicil fişini staj dönemini takip eden eğitim-öğretim döneminin başlangıcından itibaren en geç bir ay içinde ilgili bölümme imza karşılığında teslim ederler.
- (2) Staj dosyasını zamanında teslim etmeyen öğrencilerin o dönemki stajı kabul edilmez.
- (3) Bölüm staj komisyonu her eğitim-öğretim döneminde staj defterlerinin teslimini müteakip ilk 6 hafta içerisinde toplanarak staj belgelerini inceler, gerekli gördüğü takdirde öğrenciyi sözlü sınava alır, staj başarı durumunu karara bağlar ve staj sonuçları bölüm başkanlığında ilan edilir.
- (4) Öğrenciler staj değerlendirme sonuçlarına, sonuçların ilanından itibaren 7 gün içerisinde bölüm başkanlığına dilekçe ile itiraz edebilir.
- (5) Staj defterleri öğrencinin mezuniyetinden sonraki 2 yıl bölümde saklanır.

Staj Tekrarı

MADDE 11

- (1) İş yeri tarafından başarısız sayılan veya Bölüm Staj Komisyonu tarafından stajı kısmen ve/veya tamamen kabul edilmeyen öğrenci, stajını kısmen ve/veya tamamen tekrarlamak zorundadır.
- (2) Staj yeri onaylanmadan yapılan stajlar geçerli değildir.

Staj Disiplini

MADDE 12

- (1) Öğrenciler staj yaptıkları yerin disiplinine, tüzük, yönetmelik ve her türlü mevzuatına uymaya mecburdurlar.

Stajını Bitirenlerin Mezuniyet Tarihleri

MADDE 13

- (1) Bütün derslerinden başarılı olup mezun olmak için sadece stajı eksik kalan öğrencilerin,
 - a) Stajlarının bitiminde eğitim-öğretim yarıyılı başlamamışsa, staj bitirme tarihi,
 - b) Stajlarını eğitim-öğretim yarıyılı içerisinde bitirmişlerse stajın kabul edildiği tarih mezuniyet tarihi olarak kabul edilir.

Ön Lisans İçin Staj Süresi

MADDE 14

- (1) Fakülteden ön lisans belgesi alacak öğrencilerin en az 30 iş günü staj yapmış olmaları zorunludur.

Hüküm Bulunmayan Durumlar MADDE 15

- (1) Bu yönergede belirtilmemiş olan durumlar bölümlerin yapılarına ve ihtiyaçlarına göre bölüm kurullarının kararına bırakılır.

Yürürlük

MADDE 16

- (1) Bu yönerge hükümleri; Cumhuriyet Üniversitesi Senatosunun kabulünü müteakip 2015-2016 eğitim-öğretim döneminden itibaren uygulanmaya başlar.

Yürütme

MADDE 17

- (1) Bu Yönerge hükümleri, Mühendislik Fakültesi Dekanı tarafından yürütülür.

**T.C
SİVAS CUMHURİYET ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

ÖĞRENCİNİN :

Adı Soyadı :

Baba Adı :

Doğum Yeri :

Bölümü :

Sınıfı :

Okul No :

FOTOĞRAF

Yukarıda fotoğrafı yapışık ve açık kimliği yazılı bulunan kişi fakültemiz öğrencisi olup, zorunlu stajını yönetmelik gereğince yapabilir durumdadır.

Staj Yapılan Kurum :

Staj Konusu :

Staja Başlama Tarihi :

Stajı Bitirme Tarihi :

Staj Yapılan Gün Sayısı:

İşyeri-Müessesede Amirinin

Adı, Soyadı :

Unvanı :

Mühür ve İmza :

STAJ KOMİSYON RAPORU

Raporu İnceleyen Üye:

Karar:

Yapılan pratik çalışma işgünü olarak kabul edilmiştir.

.../.../20...

Bölüm Başkanı
(Adı, Soyadı, Ünvanı ve İmzası)

STAJ PROGRAMI*

**KURUM AMİRİNİN
Adı ve Soyadı
İmza ve Mühür**

*) Bu tablo staj amirince düzenlenecektir. Öğrenci staj süresi içinde Kurum Amirinin direktifleri doğrultusunda kurum içinde farklı mekân ve konularda değişik sürelerle staj yapabilirler. Bu nedenle yukarıdaki tablonun hazırlanmasında her bir çalışma konusu için işin başlama ve bitiş tarihlerinin konusunun ve kimin sorumluluğunda yapıldığının belirtilmesi gereklidir.

GÜNLÜK ÇALIŞMA PROGRAMI

S.No	Tarih	Gün	Yapılan İş
1	01/07/2024	Pazartesi	Tailwind CSS ile tasarımlar üzerinde çalışıldı. Mobil ve web için tasarımlar detaylı bir şekilde incelendi.
2	02/07/2024	Salı	Chat-app projesi üzerinde çalışıldı. Kullanılacak teknolojiler seçildi.
3	03/07/2024	Çarşamba	Chat-app projesi için login, signup ve logout işlemleri düzenlendi. JWT token kullanımı sağlandı.
4	04/07/2024	Perşembe	Mesajlaşma kısmında gönderici ve alıcı mesajları düzenlendi. Postman ile testler yapıldı.
5	05/07/2024	Cuma	Chat-app projesinin kullanıcı arayüzü (frontend) ve veritabanı yapılandırması üzerinde çalışıldı.
6	08/07/2024	Pazartesi	Tailwind CSS ve DaisyUI kullanılarak bir web tasarımı oluşturuldu.
7	09/07/2024	Salı	Chat-app projesi mesaj gönderme ve alma işlemleri için MongoDB' de tablo oluşturuldu ve backend kodları yazıldı.
8	10/07/2024	Çarşamba	Document Store projesi için veritabanı UML diyagramı tasarlandı.
9	11/07/2024	Perşembe	Document Store projesi için excalidraw üzerinden basit tasarımlar yapıldı.
10	12/07/2024	Cuma	Backend geliştirme sürecine geçildi. Veritabanı yapılandırması tamamlandı ve API entegrasyonu sağlandı.
11	16/07/2024	Salı	Veritabanı Docker bağlantısı yapıldı
12	17/07/2024	Çarşamba	Backend geliştirme devam etti. Kullanıcı kayıt ve giriş kısımları yazıldı.
13	18/07/2024	Perşembe	Document Store uygulamasının doküman post ve get istek kodları yazıldı.
14	19/07/2024	Cuma	Uygulamanın doküman silme güncelleme ve dokümanları listeleme istekleri yazıldı.
15	22/07/2024	Pazartesi	Doküman ve Kategori kısımları için kategori istekleri yazıldı.
16	23/07/2024	Salı	Backend kısımları testleri rest kullanılarak istekler test edildi.
17	24/07/2024	Çarşamba	Frontend tasarımları için renk seçimi yapıldı ve önceki excalidraw tasarım iyileştirildi.
18	25/07/2024	Perşembe	Giriş ve Kayıt ol sayfa tasarımları yapıldı.
19	26/07/2024	Cuma	Sidebar tasarımları yapıldı.
20	29/07/2024	Pazartesi	Anasayfa tasarımları yapıldı.
21	30/07/2024	Salı	Son eklenen dokümanlar için iki farklı sayfa tasarımları yapıldı.
22	31/07/2024	Çarşamba	Dokümanlar sayfası tasarımları yapıldı.
23	01/08/2024	Perşembe	Doküman yükleme sayfası tasarımları yapıldı.
24	02/08/2024	Cuma	Favori dokümanlar sayfası tasarımları yapıldı.
25	05/08/2024	Pazartesi	Backend kodları login ve sign up kısımları için flowchart çizimi yapıldı gözden kaçan kısımlara göre revize edildi.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 01/07/2024

İlk gün, şirketteki mevcut projeler ve kullanılan teknolojiler hakkında genel bir inceleme yapıldı. Şirket tarafından verilen görevler doğrultusunda Tailwind CSS ile tasarımlar üzerinde çalışıldı. Projelerin genel mimarisi, klasör yapıları, bileşen yerleşimleri ve stil dosyaları detaylı şekilde gözden geçirildi. Frontend alanında gelişim sağlamak amacıyla, yapılan tasarımlar incelendi. Hem mobil hem de web için tasarımlar detaylı bir şekilde gözden geçirildi.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 02/07/2024

Bugün şirkette, yeni açılmış olan Fontalk adlı bir platformun beta versiyonu incelemeye başlandı. İncelemenin ardından, bir mesajlaşma platformu geliştirmeye başlandı. İlk olarak, kullanılacak teknolojiler belirlendi ve MERN (MongoDB, Express, React, Node.js) stack'i tercih edildi. Bu teknolojiler arasında güçlü bir ilişki olduğu için bu seçim yapıldı.

Proje, web platformu için geliştirileceğinden, backend kısmı yazılmaya başlandı. Kullanıcı kayıt ekranı için login, signup, logout gibi POST istekleri yazıldı. Bu isteklerin kontrolü ve sonraki aşamalarda yardımcı olması amacıyla Postman kullanıldı. Projenin, SOLID prensiplerine uygun olarak yapılması için istek kısımları controllers ve routes olarak düzenlendi.

Bugün, dün başlanan chat-app projesini geliştirmeye devam edildi. Login, signup ve logout kısımları düzenlendi. Kullanıcı bilgileri, signup sırasında Postman ile kontrol edildi ve bilgilerin doğru alındığı teyit edildi. Ardından, kullanıcıya JWT token kullanarak bir oturum süresi tanımlandı. Bu sayede, kullanıcı logout işlemi yapmadıkça belirli bir süre boyunca uygulamada oturum açık kalacak.

Daha sonra, MongoDB üzerinde verileri saklamak amacıyla bir user tablosu oluşturuldu ve kullanıcı verileri bu tabloya kaydedildi. Login isteği sırasında, bu yapı yine Postman kullanılarak test edildi ve başarılı bir şekilde giriş yapılip yapılmadığı kontrol edildi. Giriş sırasında hata alındı, bu hatanın users isminin kayıtlı olup olmadığını kontrol kısmında bir sorun olduğu fark edildi ve gerekli düzenlemeler yapıldı.

```
// Login
exports.login = async (req, res) => {
  try {
    const { email, password } = req.body;

    // Gelen email e ait kullanıcı var mı
    const user = await User.findOne({ email });
    if (!user) {
      return res.status(404).json({ message: 'User not found' });
    }

    // Şifre doğruluğunu kontrol et
    const isMatch = await bcrypt.compare(password, user.password);
    if (!isMatch) {
      return res.status(400).json({ message: 'Invalid credentials' });
    }

  } catch (error) {
    res.status(500).json({ message: 'Server error' });
  }
};
```

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 04/07/2024

Bugün, mesajlaşma uygulamasında sender ve receiver kısımları için mesajların POST ve GET işlemleri yazıldı. Bu kısımda, kullanıcı bir mesaj göndermek istediginde, mesaj atacağı kişiyi seçtikten sonra, o kişinin ID'si üzerinden mesaj iletiliyor. Message tablosunda, sender ve receiver ID'leri tutuluyor ve bu sayede kullanıcılar kendi aralarında mesajlaşabilir.

Bu süreçte, atılan bir mesajı açmak istediğimde GET sorgusunun hata verdiği bir durumla karşılaşıldı. Hatanın, message tablosunda message değişkeninin yanlış yazılmasından kaynaklandığı fark edildi. Bu durum düzeltildikten sonra, mesaj atan kişinin mesajlarını sorunsuz bir şekilde görebildiği test edildi. Bu test de yine Postman kullanılarak gerçekleştirildi.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 05/07/2024

Bugün, chat-app projesinin kullanıcı arayüzü (frontend) ve veritabanı yapılandırması üzerinde çalışmalar sürdürdü. Özellikle, uygulamanın kullanıcı deneyimini iyileştirmek ve veri yönetimini daha etkili hale getirmek için çeşitli iyileştirmeler gerçekleştirildi. Kullanıcı arayüzü için, login ve signup sayfalarının görsel tasarımlarını güncellendi. Bu tasarımlar, kullanıcıların daha sezgisel ve rahat bir şekilde işlem yapabilmesi için yeniden düzenlendi. Kullanıcı dostu formlar ve hata mesajları eklendi, form doğrulama sürecini daha kullanıcı dostu hale getirildi. Formlar, kullanıcıların giriş ve kayıt işlemlerinde yaşadığı olası hataları anında gösterecek şekilde yapılandırıldı. Kullanıcı verilerinin güvenliğini artırmak için, kullanıcıların şifrelerini hash'leme ve güvenli bir şekilde saklama prosedürleri gözden geçirildi. Ayrıca, kullanıcı veritabanında performans iyileştirmeleri için indeksler eklenerek sorgu hızları artırıldı.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 08/07/2024

Bugün frontend tasarımını yapmak amacıyla Tailwind CSS ve DaisyUI kullanılarak bir web tasarımını oluşturuldu. Tasarım, login, signup ve home sayfalarından oluşmaktadır. Login kısmı için, Tailwind CSS'in basit CSS kodu yazmadan framework kullanarak detaylı bir login ve signup sayfası oluşturuldu. Bu süreçte, frontend tasarımında yeni olunduğu için etiketlerin kullanımı konusunda zorluklar yaşandı. Ancak, Tailwind CSS'in resmi dokümantasyon sitesi incelenerek bu sorunlar aşıldı.

Home page kısmında ise, sidebar kısmına bir hamburger menü oluşturuldu ve anasayfanın diğer sayfalara geçiş için şimdilik href linkleri kullanılarak geçişler sağlandı.

GÜNLÜK ÇALIŞMA PROGRAMI

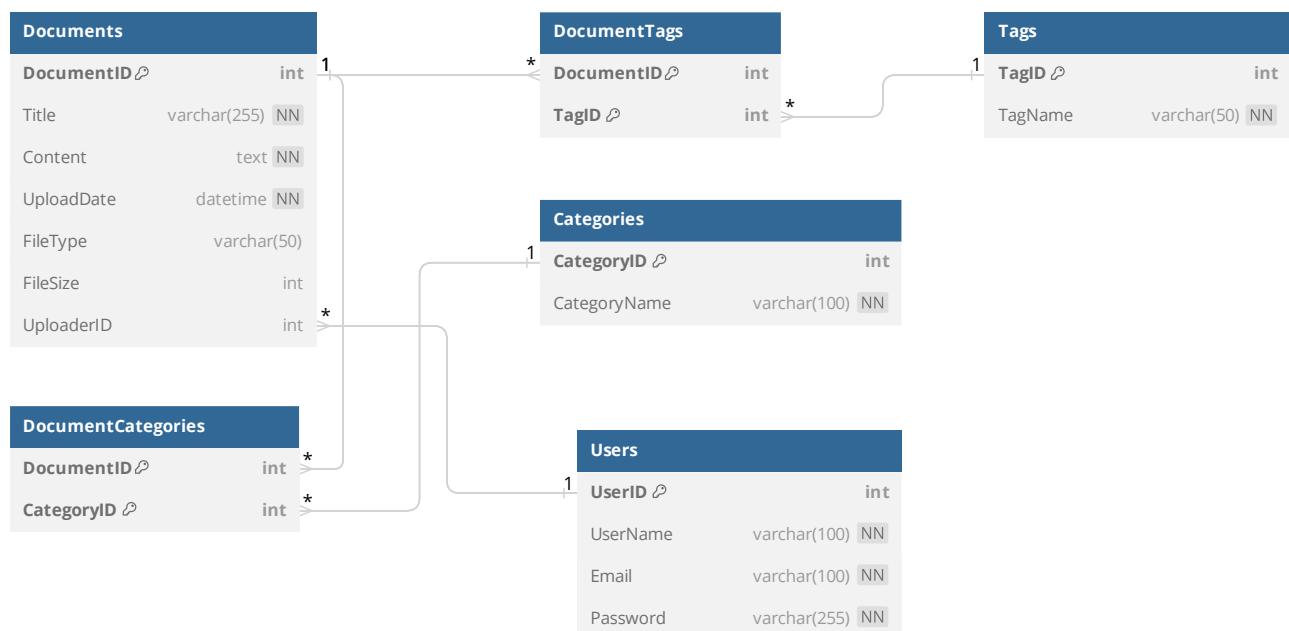
Tarih: 09/07/2024

Bugün, chat uygulaması (chat-app) projesinin son aşamalarına gelindi. Bu süreçte aşağıdaki önemli geliştirmeler ve iyileştirmeler yapıldı. Mesaj gönderen ve alan kullanıcıların ID'lerine göre verilerin MongoDB'de saklandığı bir tablo oluşturuldu. Bu tabloda, mesajın göndericisi ve alıcısının ID'leri birincil anahtar (PK) olarak belirlendi, böylece mesajların doğru şekilde ilişkilendirilmesi ve saklanması sağlandı. Bu tablo, mesajların kimin tarafından gönderildiğini ve kime gönderildiğini belirten kullanıcı ID'lerini içerir. Kullanıcılar, sol tarafta bulunan sidebar'dan seçikleri kişiye mesaj gönderebiliyor. Seçilen kişinin ID'si üzerinden mesaj gönderimi yapılmakta ve bu mesajlar MongoDB'de ilgili tabloya kaydedilmektedir. Mesaj gönderme ve alma işlevleri test edilerek düzgün çalıştığından emin olundu. Uygulamanın tüm işlevleri başarıyla tamamlandı ve proje teslim edilmek üzere hazır hale getirildi.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 10/07/2024

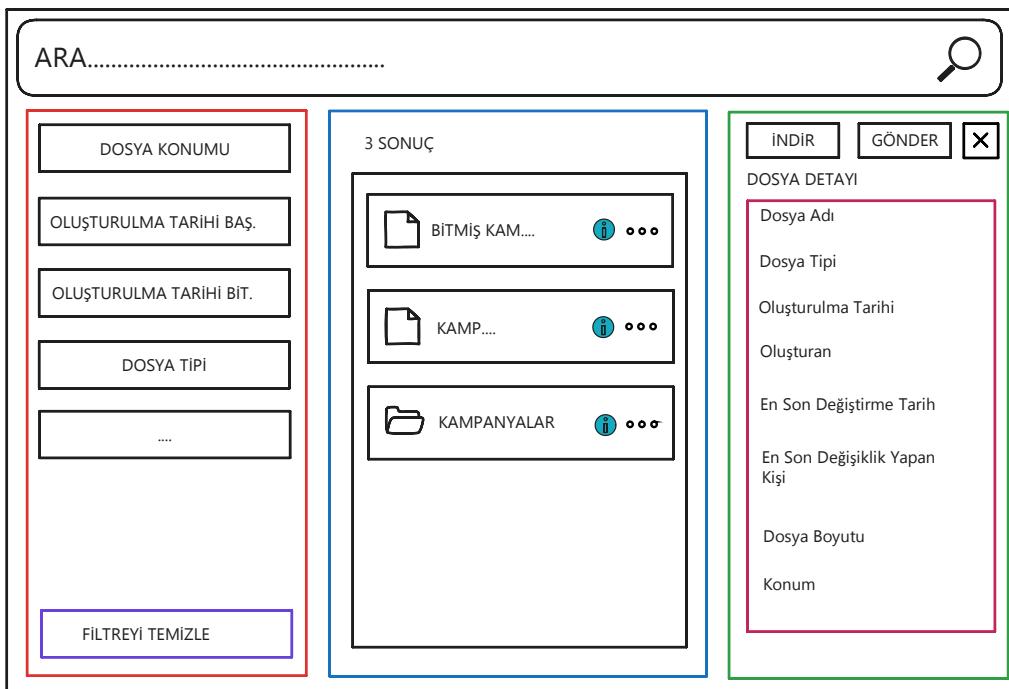
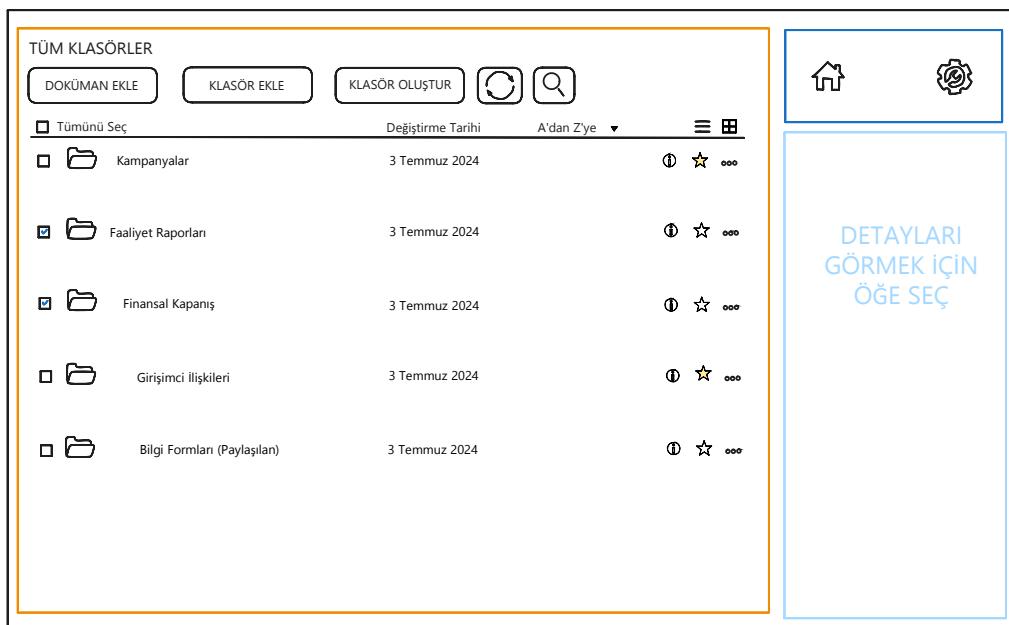
Bugün, dosya, belge ve doküman yönetimi için bir projeye başlandı. Proje kapsamında başlangıç olarak veritabanı tasarımları yapıldı. Başlangıç olarak, MSSQL için bir veritabanı gereksinim analizi gerçekleştirildi ve UML diyagramı çıkarıldı.



GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 11/07/2024

Bugün verilen görevin ikinci aşamasında, login ve signup kısımları için frontend tasarımları Excalidraw kullanılarak çizildi. Dosyaların yüklü olduğu ve kullanıcının dosyalama özelliklerini kontrol ettiği kısmın tasarımları gerçekleştirildi. Bu bölümde, dosya veya klasör yükleyerek toplu yükleme yapılabilmesi ve yeni klasör oluşturulabilmesi mümkün olacak şekilde tasarım yapıldı.



GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 12/07/2024

Bugün Veritabanını Docker kullanarak bir MSSQL image'ı oluşturuldu ve bu image'dan bir container yaratıldı. DataGrip kullanılarak veritabanına bağlanıldı ve burada tablolar oluşturuldu. Primary Key (PK), Foreign Key (FK) bağlantıları ve ilişkiler dikkate alınarak tablolar düzenlendi.

```
CREATE TABLE Users (
    UserID int AUTO_INCREMENT PRIMARY KEY,
    UserName varchar(100) NOT NULL,
    Email varchar(100) NOT NULL UNIQUE,
    Password varchar(255) NOT NULL,
    RegistrationDate datetime NOT NULL);
```

```
CREATE TABLE Documents (
    DocumentID int AUTO_INCREMENT PRIMARY KEY,
    Title varchar(255) NOT NULL,
    Content text NOT NULL,
    UploadDate datetime NOT NULL,
    FileType varchar(50),
    FileSize int,
    UploaderID int,
    FOREIGN KEY (UploaderID) REFERENCES Users(UserID));
```

```
CREATE TABLE Tags (
    TagID int AUTO_INCREMENT PRIMARY KEY,
    TagName varchar(50) NOT NULL);
```

```
CREATE TABLE DocumentTags (
    DocumentID int,
    TagID int,
    PRIMARY KEY (DocumentID, TagID),
    FOREIGN KEY (DocumentID) REFERENCES Documents(DocumentID),
    FOREIGN KEY (TagID) REFERENCES Tags(TagID));
```

```
CREATE TABLE Categories (
    CategoryID int AUTO_INCREMENT PRIMARY KEY,
    CategoryName varchar(100) NOT NULL);
```

```
CREATE TABLE DocumentCategories (
    DocumentID int,
    CategoryID int,
    PRIMARY KEY (DocumentID, CategoryID),
    FOREIGN KEY (DocumentID) REFERENCES Documents(DocumentID),
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID));
```

Bugün, projede dün oluşturulan docker da ki veritabanına bağlanmak için JavaScript dili ve Node.js kullanılarak bağlantı sağlandı. Docker üzerinde MSSQL çalıştırıldığı için, bağlantı kısmının nasıl yapılacağı hakkında araştırma yapmak gerekti. Bu süreçte, "server.js" adında bir dosya oluşturuldu. Bu kısımda gerekli konfigürasyonlar yer aldı.

"db.js" dosyasında veritabanı bağlantısını gerçekleştirmek için YouTube ve Stack Overflow'dan birkaç saat süren bir araştırma yapıldı ve bağlantı sağlandı. Bu aşamada yaşanan sorunlar ve çözüm yöntemleri detaylı olarak incelendi.

```
const express = require('express');
const dotenv = require('dotenv');
const { Sequelize } = require('./database/db'); // sql modülünü de içe aktarıyoruz

// Ortam değişkenlerini yükleme
dotenv.config();
const PORT = process.env.PORT || 4343;
const app = express();

// Middleware
app.use(express.json());

// Routes
require('./startup/routes')(app);

// Database connection
Sequelize.authenticate()
.then(() => {
  console.log('Database connection successful');
  app.listen(PORT, () => console.log(`Server is running on port ${PORT}`));
})
.catch(err => {
  console.error('Unable to connect to the database:', err);
});
```

Bugün, projenin login ve signup kısımları için Sequelize kullanıldı. İlk olarak, kullanıcı oluşturma işlemi için Create adında bir POST sorgusu yazıldı. Bu sorguda, kullanıcının email ve password değişkenleri dikkate alındı. Aynı email adresinde bir başka kullanıcı kaydına izin verilmedi ve bu durum için hata mesajı gönderildi. Şifre değişkeni ise, npm bcrypt kütüphanesi kullanılarak hash'lendi. Bu yöntemle, kullanıcının şifresinin veritabanında açık olarak görünmesi engellendi. Tüm bu işlemler, frontend olmadan test.rest adında bir dosya ile test edildi. Login kısmından sonra, signup işlemi için, kullanıcının giriş yaparken Sequelize'da findOne metodu ile email kaydının var olup olmadığı kontrol edildi. Sonrasında, bcrypt'in compare metodu kullanılarak girilen şifrenin, veritabanında kayıtlı şifre ile eşleşip eşleşmediği kontrol edildi.

```
// User create
const createUser = async (req, res) => {
  const { UserName, LastName, Email, Password } = req.body;

  try {
    // Kayıtlı email kontrolü
    const existingUser = await User.findOne({ where: { Email } });

    if (existingUser) {
      return res.status(400).send({ message: "Email already exists" });
    }

    // Şifre hashleme
    const saltRounds = 10;
    const hashedPassword = await bcrypt.hash(Password, saltRounds);

    const user = await User.create({ UserName, LastName, Email, Password: hashedPassword });
    res.status(200).send({ message: "Successfully added user", user });

  } catch (err) {
    console.log('User create error:', err);
    res.sendStatus(500);
  }
}
```

Bugün, Docstore (dosya arşivi) uygulamasının doküman kısmının backend kodları yazıldı. Bu kısım, iki adet GET ve üç adet POST isteğinden oluşmaktadır:

GET İstekleri:

- 1- GET isteği, tüm dokümanları listelemektedir.
- 2- GET isteği, sadece seçilen dokümana ait bilgileri listelemektedir.

POST İstekleri:

- 1- POST isteği, yeni bir doküman eklemek için kullanılmaktadır. Dokümanın bilgileri (örneğin, başlık gibi) veritabanına gönderilmektedir. Bu işlem, REST kullanılarak test edilmiştir.
- 2- POST isteği, mevcut dokümanın bilgilerini güncellemektedir. Güncelleme işlemi için dokümanın ID'si (documentId) kullanılmaktadır. Bu ID, kullanıcıların seçikleri dokümani güncelleyebilmeleri için önemlidir ve doğru ID kullanılmadığında sorun yaşanabilir.
- 3- POST isteği, seçilen dokümanı silmektedir. Bu işlemde de ID kullanılarak, seçilen doküman veritabanından silinmektedir.

Tüm bu işlemler, REST kullanılarak test edildi ve ilgili fonksiyonlar kontrol edildi.

Doküman Listeleme

GET <http://localhost:4343/document/list>
Content-Type: application/json

Seçilen Dokümanı listeleme

GET
<http://localhost:4343/document/detail/24>
Accept: : application/json

Doküman Ekleme

POST <http://localhost:4343/document/create>
Content-Type: application/json

```
{  
    "Title": "test1",  
    "Content": "yazılım categorisi ekleme ",  
    "FileType": "pdf",  
    "FileSize": 102000,  
    "UploaderID": ,  
    "FileURL": "plans/project_plan_q3.xlsx",  
}
```

Doküman Ekleme

POST <http://localhost:4343/document/create>
Content-Type: application/json

```
{  
    "Title": "test1",  
    "Content": "yazılım categorisi ekleme ",  
    "FileType": "pdf",  
    "FileSize": 102000,  
    "UploaderID": ,  
    "FileURL": "plans/project_plan_q3.xlsx",  
}
```

Doküman Güncelleme

POST <http://localhost:4343/document/update/21>
Content-Type: application/json

```
{  
    "Title": "Project Plan (değiştirdim)",  
    "Content": "Detailed project plan for Q3",  
    "FileType": "xldasdas",  
    "FileSize": 1024000,  
    "UploaderID": 1,  
    "FileURL": "değiştirildi"  
}
```

Doküman Silme

POST <http://localhost:4343/document/delete/23>
Content-Type: application/json

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 19/07/2024

Bugün, dokümanların düzenli bir şekilde yönetilebilmesi için bir category tablosu oluşturuldu. Bu tabloda, kategori oluşturma, silme ve güncelleme işlemleri için gerekli sorgular yazıldı.

Veritabanında category ve document tabloları arasında bir ilişki kuruldu. Bu ilişki, documentCategory tablosu aracılığıyla sağlanmaktadır. Bu yapı sayesinde, bir doküman birden fazla kategoriye ait olabilir ve bir kategori birden fazla doküman içerebilir. Bu düzenleme, dokümanların daha iyi bir şekilde organize edilmesini ve daha kolay bulunmasını sağlayacaktır.

Bugün, documentCategory oluşturmak için bir POST isteği yazıldı. Uygulamanın içerisinde her bir klasörün veya PDF'in bir kategorisi olması gerektiğinden, bu düzenleme ile daha düzenli bir listeleme ve arama yapılabilir hale geldi. Ayrıca, kategori listeleme kısmı geliştirildi. Bu işlem, document, category ve documentCategory tabloları kullanılarak gerçekleştirilmiştir. Sorgu sonucunda, hangi dosyanın hangi kategoride olduğu bilgisi döndürülmektedir. Bu özellik, frontend tarafında kullanılacak ve işin önemli bir parçasını oluşturacaktır. Son olarak, kullanılmayan kategorileri silmek için bir DELETE isteği yazıldı. Bu istek, gereksiz kategori verilerini veritabanından temizleyerek sistemin düzenli ve verimli çalışmasını sağlayacaktır.

```
// Dokümanın kategorilerini getir
const getDocCategories = async (req, res) => {
  const documentId = parseInt(req.params.documentId, 10);

  if (isNaN(documentId)) {
    return res.status(400).send({ message: "Invalid document ID" });
  }

  try {
    const docCategories = await DocumentCategory.findAll({
      where: { DocumentID: documentId },
      include: [
        {
          model: Category,
          attributes: ['CategoryName']
        }
      ],
    });

    // sorgu ile veritabanından dönen sonucu düzenle
    const result = docCategories.map(dc => ({
      DocumentID: dc.DocumentID,
      CategoryID: dc.CategoryID,
      CategoryName: dc.Category.CategoryName
    }));

    res.status(200).send(result);

  } catch (error) {
    console.log('Document category get error:', error);
    res.sendStatus(500);
  }
};
```

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 23/07/2024

Backend kısımlarının sorguları tamamlandığı için frontend tasarımlarına geçildi. Projenin başında temel bir tasarım oluşturmak için Excalidraw kullanılmıştı, ancak daha görsel ve profesyonel bir tasarım için Figma kullanımı öğrenildi. Önceden tasarım deneyimi olmadığı için bu süreçte zorluk yaşandı. Ancak yapılan tasarımları incelediğinde, daha sade ve abartıdan uzak bir tasarım oluşturulmaya başlandı. Bu aşamada, login ve signup kısımları için tasarımlar hazırlandı. Görsel ve detay eklemeleri de yapıldı. Tasarımın sade ve kullanıcı dostu olmasına özen gösterildi.

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih: 24/07/2024

Bugün, anasayfa tasarımını yapmak için Figma derslerini izlendi bilgi edinildi. Frame oluşturma ve bir tasarımda nelere dikkat edilmesi gerektiği öğrenildi. Renk çemberi kullanılarak renk tanımı yapıldı; tamamlayıcı renkler seçildi. Seçilen renkler, 3 tane aynı bölgede renk, 2 tane karşıt renk ve 2 tane 45 derecelik açı oluşturacak şekilde pastel tonlarda belirlendi. Bu seçim, en iyi renk uyumunu sağlamak amacıyla yapıldı. Daha sonra, anasayfanın tasarımını önce kağıt üzerinde çizerek planlama yapıldı. Bu taslak, Figma üzerinde yapılacak olan tasarımın temelini oluşturdu.

Bugün, uygulamanın kayıt ve authentication (auth) kısmı üzerinde çalışıldı. Login ve signup tasarımları yapıldı. Login kısmında iki adet box, signup kısmında ise dört adet box kullanıldı. Box'lar, email, ad, soyad ve password gibi alanları içerir.

Ayrıca, Figma derslerini tekrar izleyerek, bir tasarımda yazıların görünürlüğü ve kontrast ayarlarının nasıl olması gereği hakkında bilgi edinildi. Kullanıcı deneyimi (UI/UX) için en iyi uygulama yöntemleri öğrenildi.

The image displays two screenshots of a mobile application interface. Both screens feature a red logo with a white 'fp' monogram and the text 'Document Store'.

Top Screenshot (Login Screen):

- Title:** GİRİŞ YAP
- Fields:** E-mail (with an envelope icon), Şifre (with a lock icon), and a checkbox labeled 'Şifremi Unuttum'.
- Buttons:** A blue 'GİRİŞ YAP' button at the bottom.

Bottom Screenshot (Sign-up Screen):

- Title:** KAYIT OL
- Fields:** Ad (with a person icon), Soyad (with a person icon), E-mail (with an envelope icon), and Şifre (with a lock icon).
- Buttons:** A blue 'KAYIT OL' button at the bottom.

GÜNLÜK ÇALIŞMA PROGRAMI

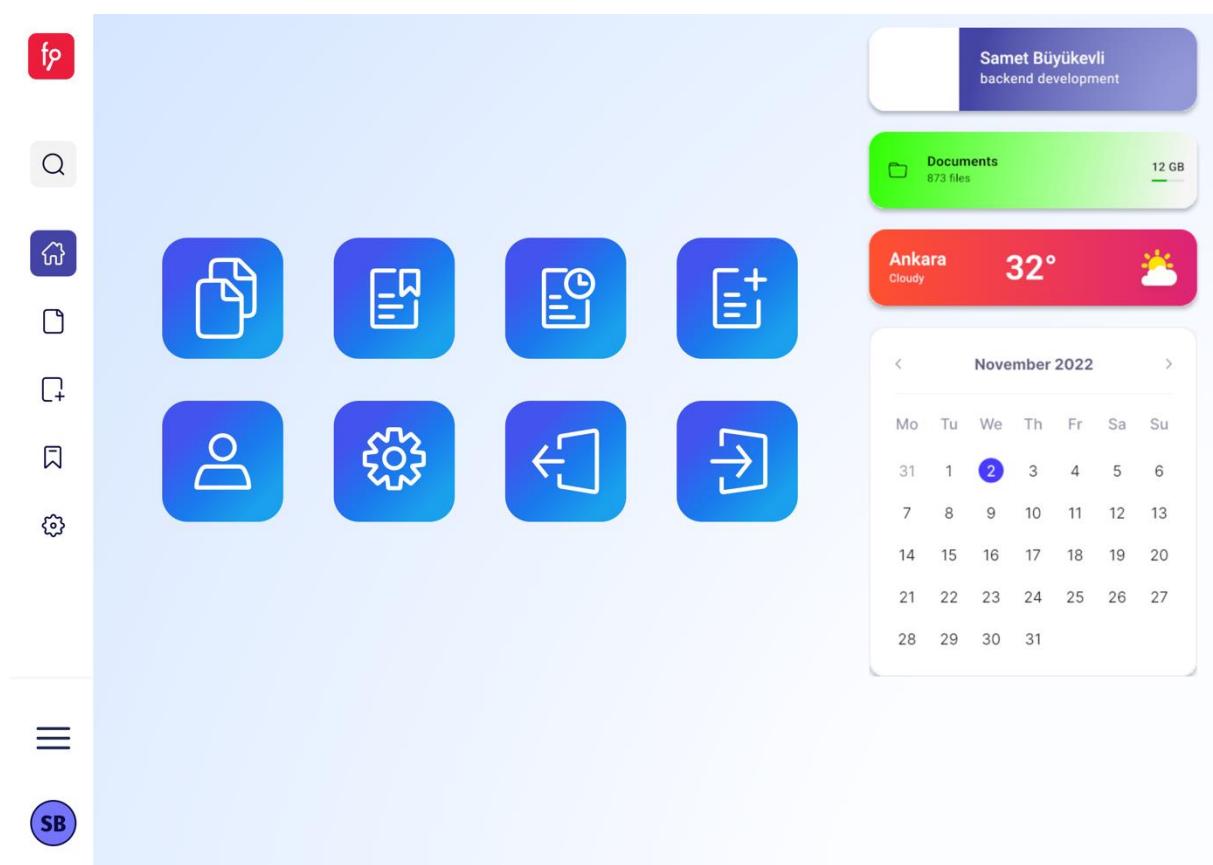
Tarih: 26/07/2024

Bugün, anasayfa tasarımını dün kağıt üzerinde çizdiğim taslağa göre oluşturmaya başladım. İlk olarak bir sidebar ekledim. Sidebar içerisinde, "Home", "My Document", "Add Document", "Favourites" ve "Settings" ikonları ve metinleri yerleştirdim. Bu kısım, kullanıcının sayfalar arasında geçiş yapabilmesi için bir navigasyon görevi görecektir.

Bugün, anasayfa tasarımlı widget kullanımı üzerine çalışıldı. Widget'ların nasıl dizayn edileceği ve ne tür tasarımlar yapılması gerekiği hakkında bilgi edinmek için birkaç video izlendi.

Sonrasında, hava durumu ve takvim widget'larını yapmaya başladım. Özellikle takvim widget'ını oluşturmak zorlayıcı oldu; hazır bir takvim frame'ini kullanmak yerine kendi tasarımımı yapmayı tercih ettim. Her sayının ve satır ile sütunlar arasındaki eşit aralıkları ayarlamak biraz zorlayıcıydı, ancak başarılı bir şekilde tamamlandı.

Ayrıca, kullanıcı fotoğrafı ve ismini içeren bir kart tasarımlı yaptım. Bu kartta, uygulamada kaç tane doküman olduğu ve ne kadar GB depolama alanı kapladığı bilgilerini gösteren bir widget oluşturuldu. Depolama alanı göstergesinin renk kısmını diğer widget'lardan farklı olarak linear bir şekilde tasarlandı ve depolamanın dolu olduğunu belirten bir efekt oluşturuldu.



Bugün, widget ile geçilen "Recently Documents" kısmının tasarımını yaptım. Bu bölümde dosyalar iki farklı şekilde görüntülenebilir: bir liste şeklinde ve yatay bir satırda. Her iki tasarım da gerçekleştirildi.

Bir dosya aşağıdaki bilgileri içeriyor:

- Icon: Dosya türünü belirtir.
- Ad: Dosyanın adı.
- Yükleme Tarihi: Dosyanın yüklentiği tarih.
- Favourites İkonu: Dosyanın önem sırasına göre renkleri olan kırmızı, mavi, sarı, yeşil, mor gibi ikonlar içerir.
- Info İkonu: Bu ikona tıklanlığında dosya hakkında daha detaylı bilgiler gösterilir.
- Yıldız İkonu: Kullanıcıların belgeleri yıldızlayarak anasayfada öncelikli olarak görünmesini sağlar.
- Üç Nokta (More Options): Bu ikona tıklanlığında dosya silme veya diğer görevler gerçekleştirilebilir.

The screenshot shows the 'Recently Added Documents' section of the 'Document Store' application. On the left, there's a sidebar with navigation links: Home (selected), My Documents, Add Document, Favourites, Settings, and a user profile for 'Samet Büyükevli'. The main area displays a grid of recently added documents. Each document item includes a small icon representing the file type (PDF, DOCX, RAR, XLS), the document name, its creation date, and a set of interactive icons: a bookmark, three dots, an info icon, a star, and a more options icon. The stars are color-coded (red, green, blue, yellow) based on their importance.

Document Type	Name	Date	Importance (Star Color)
PDF	Kampanyalar.pdf	06/08/2024 12:15	Yellow
DOCX	FaaliyetRaporu.docx	05/08/2024 14:15	Yellow
RAR	Kampanyalar.rar	06/08/2024 12:15	Yellow
XLS	ihraçEdilenŞirketler.xls	06/08/2024 12:15	Yellow
PDF	Kampanyalar.pdf	06/08/2024 12:15	Yellow
DOCX	FaaliyetRaporu.docx	05/08/2024 14:15	Yellow
RAR	Kampanyalar.rar	06/08/2024 12:15	Yellow
XLS	ihraçEdilenŞirketler.xls	06/08/2024 12:15	Yellow
PDF	Kampanyalar.pdf	06/08/2024 12:15	Yellow
DOCX	FaaliyetRaporu.docx	05/08/2024 14:15	Yellow
RAR	Kampanyalar.rar	06/08/2024 12:15	Yellow
XLS	ihraçEdilenŞirketler.xls	06/08/2024 12:15	Yellow
PDF	Kampanyalar.pdf	06/08/2024 12:15	Yellow
DOCX	FaaliyetRaporu.docx	05/08/2024 14:15	Yellow
RAR	Kampanyalar.rar	06/08/2024 12:15	Yellow
XLS	ihraçEdilenŞirketler.xls	06/08/2024 12:15	Yellow

Bugün, şirkette "My Documents" sayfasının arayüz tasarımları yapıldı. Kullanıcının eklediği dokümanları belirli özelliklere göre sıralayabilmesi için çeşitli özellikler eklendi. Ayrıca, kullanıcıların istedikleri dokümanı arayabilmesi için bir arama çubuğu yerleştirildi. Arama çubuğu, kullanıcıların dokümanı datetime, code veya ad gibi kriterlere göre aramasını sağlar.

- Datetime: Dokümanın yükleniği tarih ve saat bilgisi yyyyMMddHHmm formatında tutuluyor, bu sayede arama işlemi kolaylaştırır.
 - Yıldız İkonu: Her doküman için bir yıldız ekleyerek dokümanların öne çıkmasını sağlar.
 - Flag: Her dokümanın bir flag'i bulunuyor, bu da dokümanların önem sırasına göre farklı renklerle kategorize edilmesini sağlar.

Ek olarak, kullanıcılar istedikleri bir veya birden fazla dokümanı transfer butonu ile belirttikleri e-posta adresine gönderir. Ayrıca, dokümanları silmek için "Delete" butonunu kullanarak toplu veya seçilen dokümanları silebilir.

Documents		<input type="button" value="Upload"/>	<input type="button" value="Share"/>	<input type="button" value="Edit"/>	<input type="text" value="Search for a Document..."/>	
<input type="checkbox"/>	Name <input type="button" value="Sort"/>	Date-Time <input type="button" value="Sort"/>	Code <input type="button" value="Sort"/>	Date Modified	File Size <input type="button" value="Sort"/>	<input type="button" value="Delete"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
<input type="checkbox"/>	 FaaliyetRaporu.pdf	202408061415_	06/08/2024 14:15	5 MB	<input type="button" value="More"/>   	<input type="button" value="Details"/>
 Samet Büyükevli						

Bugün, "Add Document" sayfasının tasarımını gerçekleştirdim. Bu tasarım, kullanıcıların tek bir veya birden fazla doküman yüklemelerini sağlayacak şekilde hazırlandı.

1. Tek Dosya Yükleme: Kullanıcı bir dosya yüklediğinde, dosya otomatik olarak "Upload Doc" kısmına ekleniyor. Bu kısımda dosya bilgileri otomatik olarak dolduruluyor:
 - Name: Dosyanın adı
 - Date Time: Dosyanın yükleme tarihi ve saatı
 - Code: Dosya ile ilgili özel bir kod
 - Doc Size: Dosyanın boyutu
 - Doc Type: Dosya türü
 - Explanation: Dosyanın açıklaması
2. Manuel Düzenleme: Kullanıcı, bu alanları manuel olarak değiştirebilir veya ekleyebilir. Ek olarak:
 - Upload Date: Yükleme tarihi
 - Tags: Kullanıcılar, # işaretleri ile etiketler oluşturabilir, bu sayede dokümanlar daha iyi kategorize edilebilir ve arama yaparken kolaylık sağlar.

The screenshot shows the 'Add Document' page of the 'Document Store' application. On the left, there's a sidebar with a logo, a search bar, and links for Home, My Documents, Add Document (which is highlighted in blue), Favourites, Settings, and a user profile for 'Samet Büyükevli'. The main area has a large 'Upload Document' button with an icon. Below it are several input fields: 'Name' (with a capital letter 'A'), 'Date-Time Code' (with a calendar icon), 'Tags' (with a '#' symbol), 'Upload Date' (with a date picker icon), 'Document Size' (with a megabyte icon), and 'Document Type' (with a file icon). To the right of these fields is a 'Preview' section. At the bottom is an 'Explanation' text area and a 'Logout' link.

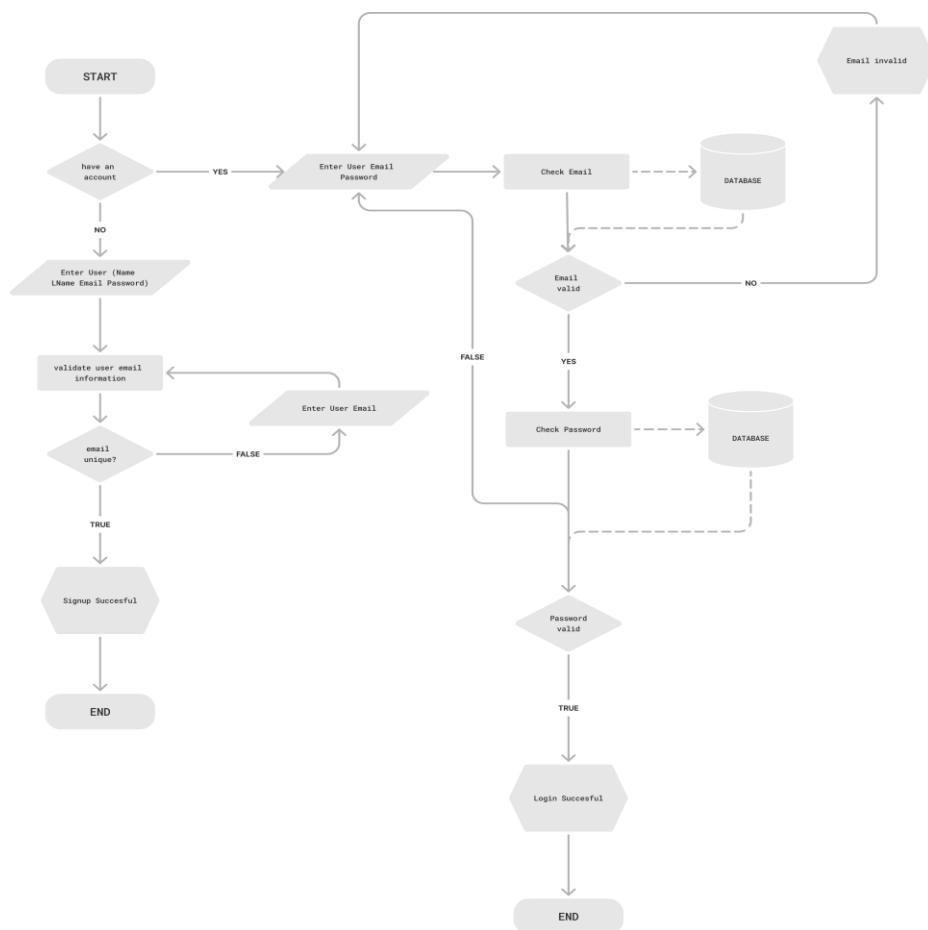
Bugün, "Favorites" kısmının tasarımlarını yaptım. Arama kısmında kullanıcıların flag'ler seçebilmesini sağlayan bir özellik eklendi. Kullanıcılar, mevcut renk seçeneklerine göre flag'leri seçerek dokümanları filtreleyebilir.

- Flags Seçimi: Arama kısmında farklı renkteki flag'leri seçme imkanı sağlandı.
 - Custom Flags Oluşturma: Kullanıcılar, "Settings" kısmına giderek özel flag'ler oluşturabilirler. Burada kullanıcılar, istedikleri renkte ve isimde flag oluşturabilirler. Bu özellik, dokümanların önem sırasına göre düzenlenmesini ve aranan dokümanın bulunmasını hızlandırdı.

The screenshot shows a document management application interface. On the left, there's a sidebar with navigation links: Home, My Documents, Add Document, Favourites (which is currently selected and highlighted in blue), Settings, and a user profile for 'Samet Büyükevli'. The main area is titled 'Favourites' and lists 12 documents, all named 'FooliyetRaporu.pdf', each with a creation date of '2024/08/06 14:15', a size of '5 MB', and a download icon. To the right of the list is a search bar for 'Search flags...' and a dropdown menu for selecting document colors: Red, Blue, Green, and Plus. The 'Red' option is currently selected. Below the dropdown are buttons for 'Select All', 'Yellow', and 'Purple'.

Bugün, kodlama kısmına geçmeden önce mevcut backend kodlarını inceledi. Dosya ekleme, silme, güncelleme, login ve signup işlemlerinin temel olarak çalıştığı görüldü. Ancak, dosyaların kimlerin erişimine açık olacağı, kimlerin değiştirebileceği veya silebileceği gibi erişim kontrollerinin eksik olduğunu fark edildi.

- Erişim Kontrolleri: Backend kodlarının düzenlenmesi için yöneticimin önerisi üzerine, bir iş akış diyagramı (flowchart) çizildi
- Authentication Tasarımı: Tasarım sırasında, kullanıcıların e-posta hatası aldıkları istedikleri kadar deneme yapabildiklerini fark edildi. Kullanıcıyı bu durumda döngüden çıkarmak için, 3 başarısız denemeden sonra kullanıcıyı belirli bir süre sisteme giriş yapamayacak şekilde tasarlandı. Bu özellik, sistem güvenliğini artırmak için önemli bir değişiklikti. Bu kısımda bir "loop limit" ekleyerek kodda gerekli düzenlemeleri yapıldı.



GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

GÜNLÜK ÇALIŞMA PROGRAMI

Tarih:/...../ 20.....

