FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği Bölümü

YMH459 Yazılım Mühendisliği Güncel Konular
Proje Dokümantasyonu

Güvenli Bulut Depolama Projesi

Dijital Safe Box Web Sitesi

Yazan
180541301 - Yunus Emre ES

Proje Yürütücüleri

Doç. Dr. Fatih ÖZKAYNAK
Arş. Gör. Vahtettin Cem BAYDOĞAN

OCAK – 2022

1. **PROJE ADI**

Dijital Safe Box (Güvenli Bulut Depolama Projesi)

2. PROJE AMACI

Günümüzde birçok işlemi internet üzerinden dijital ortamlarımızda yapmaktayız. Fotoğraflarımız, dökümanlarımız ve bizim için önemli olan çoğu verimiz bu dijital ortamlarda siber saldırılara maruz kalabilmektedir. Gelişen teknoloji ile verilerimizi depolamak ne kadar önemli ise, verilerimizin güvenliğini de sağlamak bir o kadar önemlidir. Bu hedefle hayata geçirilen Dijital Safe Box web sitemiz ile verilerimizi ister kendi depolama alanınızda ister hizmet sunduğumuz dijital kasalarımızda kendi belirlediğiniz şifreler ile AES ve Fernet Algoritma şifreleme teknikleriyle güvenle saklayabilmektesiniz.

3. PROJE EKİBİ

- Proje Yöneticisi : Yunus Emre ES
- Web site tasarım işlemleri : Yunus Emre ES
- Veritabanı tasarım işlemleri : Yunus Emre ES
- Yazılım ve Kod düzenleme işlemleri : Yunus Emre ES
- Proje dökümantasyon işlemleri : Yunus Emre ES

4. PROJEDE KULLANILAN YAZILIMSAL BİLEŞENLER

Python

Python, nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir. [1]

Projemizin tasarlanmasında Python dili ve Python dilinde yazılmış kütüphaneler kullanılmıştır.

Proje Kaynak dosyalarına ulaşmak için:
https://github.com/xsmileh/ymgk_digitalsafebox

HTML

Hiper Metin İşaretleme Dili (İngilizce Hypertext Markup Language, ks. HTML) web sayfalarını oluşturmak için kullanılan standart metin işaretleme dilidir. Dilin son sürümü HTML5'tir. [2]

Projemizin web sayfalarının oluşturulmasında kullanılan yazılım dilidir.

CSS

Cascading Style Sheets (Basamaklı Stil Şablonları ya da Basamaklı Biçim Sayfaları, bilinen kısa adıyla CSS), HTML'e ek olarak metin ve format biçimlendirme alanında fazladan olanaklar sunan bir işaretleme dilidir. [3]

Projemizin sayfa tasarım düzenlemelerinde kullanılan ve ./static isimli klasör içerinde bulunan .css uzantılı yazılım dilidir. Boostrap kütüphaneleri ile birlikte kullanılmıştır.

Javascript

JavaScript, yaygın olarak web tarayıcılarında kullanılmakta olan dinamik bir programlama dilidir. JavaScript ile yazılan istemci tarafı betikler sayesinde tarayıcının kullanıcıyla etkileşimde bulunması, tarayıcının kontrol edilmesi, asenkron bir şekilde sunucu ile iletişime geçilmesi ve web sayfası içeriğinin değiştirilmesi gibi işlevler sağlanır. [4]

Projemizin tasarım ve upload alanlarında dinamik geçişlerle ilgili çalışmalarda kullanılmıştır. ./static isimli klasör içerinde bulunan .js uzantılı yazılım dilidir.

Sublime Text

Sublime Text, içinde birçok programlama dili arayüzü barındıran, çapraz platform bir kaynak kod düzenleme ve metin editörüdür. Arayüzü Vim'den ilham alınarak tasarlanmıştır. Sublime-paketleri (Sublime-packages) yardımıyla fonksiyonelitesi genişletilebilir ancak Sublime Text açık kaynaklı ya da özgür bir yazılım değildir. Buna rağmen genişleme paketlerinin pek çoğu özgür yazılım lisansı ile dağıtılmakta ve Sublime Text kullanıcılarının oluşturduğu topluluk tarafından geliştirilmektedir. [5]

Projemizin tasarlanmasında kullanılan metin editörüdür. Python dosyalarının çalıştırılmasında ve web dosyalarının düzenlenmesinde hızlı ve efektif çözümler için projeye dahil edilmiştir.

Boostrap Kütüphanesi

Twitter Bootstrap (ya da kısaca Bootstrap) açık kaynak kodlu, web sayfaları veya uygulamaları geliştirmek için kullanılabilecek araçlar bütünü ve önyüz çatısı. Bootstrap, web sayfaları veya uygulamalarında kullanılabilecek, HTML ve CSS tabanlı tasarım şablonlarını içerir. Bu şablonlar form, navigasyon çubuğu, buton gibi arayüz bileşenleri oluşturmakta kullanılabilmektedir. Ocak 2021 itibarı ile Bootstrap, Github üzerinde 148 binin üzerinde "star" ile 71 binin üzerinde "fork" sayılarına ulaşarak, sitenin en popüler projelerinden biri olmuştur. [6]

Projemiz web sitesi tasarımında kullanılmıştır.

Flask Kütüphanesi

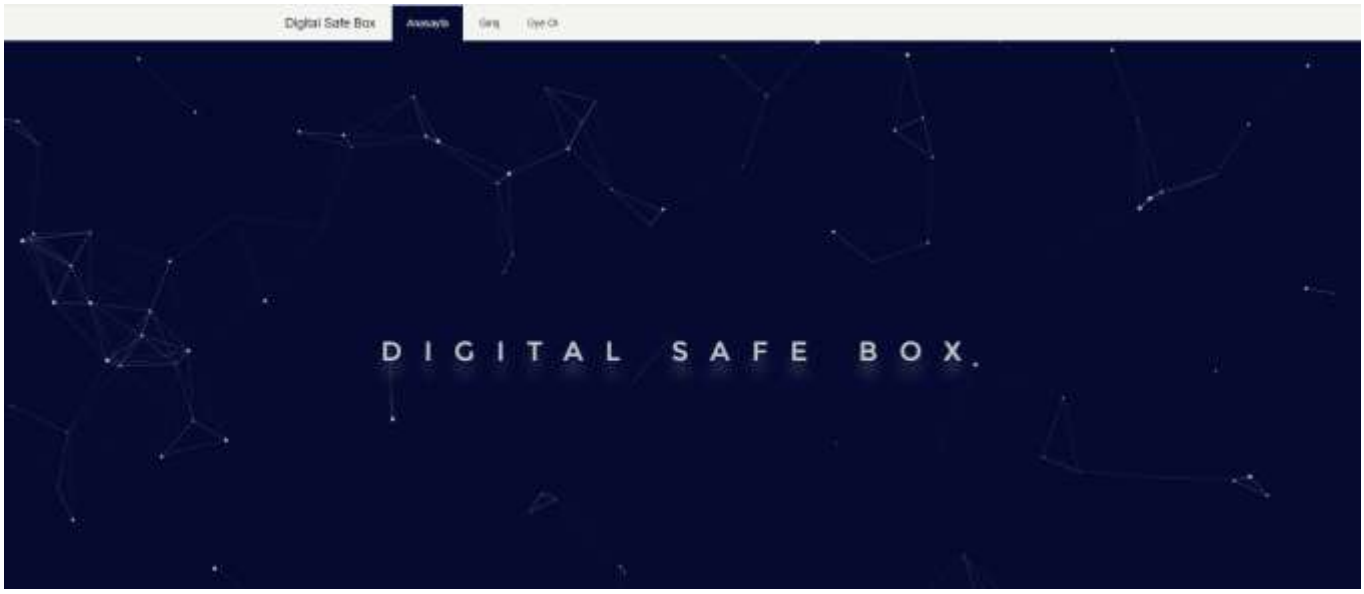
Flask, Python ile yazılmış bir mikro web çerçevesidir. Belirli araçlar veya kitaplıklar gerektirmediğinden mikro çerçeve olarak sınıflandırılır. Veritabanı soyutlama katmanı, form doğrulama veya önceden var olan üçüncü taraf kitaplıklarının ortak işlevler sağladığı diğer bileşenlere sahip değildir. Ancak Flask, Flask'ın kendisinde uygulanmış gibi uygulama özellikleri ekleyebilen uzantıları destekler. Nesne-ilişkisel eşleyiciler, form doğrulama, karşıya yükleme işleme, çeşitli açık kimlik doğrulama teknolojileri ve çeşitli ortak çerçeve ile ilgili araçlar için uzantılar mevcuttur. [7]

Projemizin veritabanı ve web site elementleri arasında geçişlerde Python dili ile bağlantısında kullanılmıştır.

Fernet Crytography	Dosyalarımızın veya şifrelerimizin şifrelenmesi işlenminde kullanılan Python dili ile yazılmış kütüphanemizdir. [8] 32 bit şifreleme yapan Fernet algoritması ile Proje kapsamında önce verilerimiz şifrelenmiştir.
AES Crytography	AES (Advanced Encryption Standard; Gelişmiş Şifreleme Standardı), elektronik verinin şifrelenmesi için sunulan bir standarttır. Amerikan hükûmeti tarafından kabul edilen AES, uluslararası alanda da defacto şifreleme (kripto) standardı olarak kullanılmaktadır. DES'in (Data Encryption Standard - Veri Şifreleme Standardı) yerini almıştır. AES ile tanımlanan şifreleme algoritması, hem şifreleme hem de şifreli metni çözmede kullanılan anahtarların birbiriyle ilişkili olduğu, simetrik-anahtarlı bir algoritmadır. AES için şifreleme ve şifre çözme anahtarları aynıdır. [9] Projemiz kapsamında Fernet ile şifrelediğimiz verilerimizi kullanıcıdan alınan Şifre bilgisi ile 128 bit olarak şifreleme tekniği kullanarak şifrelenmesinde kullanılmıştır.
SQLAlchemy	SQLAlchemy, MIT Lisansı altında yayınlanan Python programlama dili için açık kaynaklı bir SQL araç takımı ve nesne-ilişkisel eşleyicidir. [10] Projemiz kapsamında kullanıcı bilgileri ve dosya bilgilerinin veritabanına kaydedilmesi ve veritabanındaki veriler ile web sitesi arasındaki iletişim için kullanılan Python kütüphanesidir.
Diğer kütüphaneler	os, datetime, request, send_file, socket, md5, b64encode, b64decode

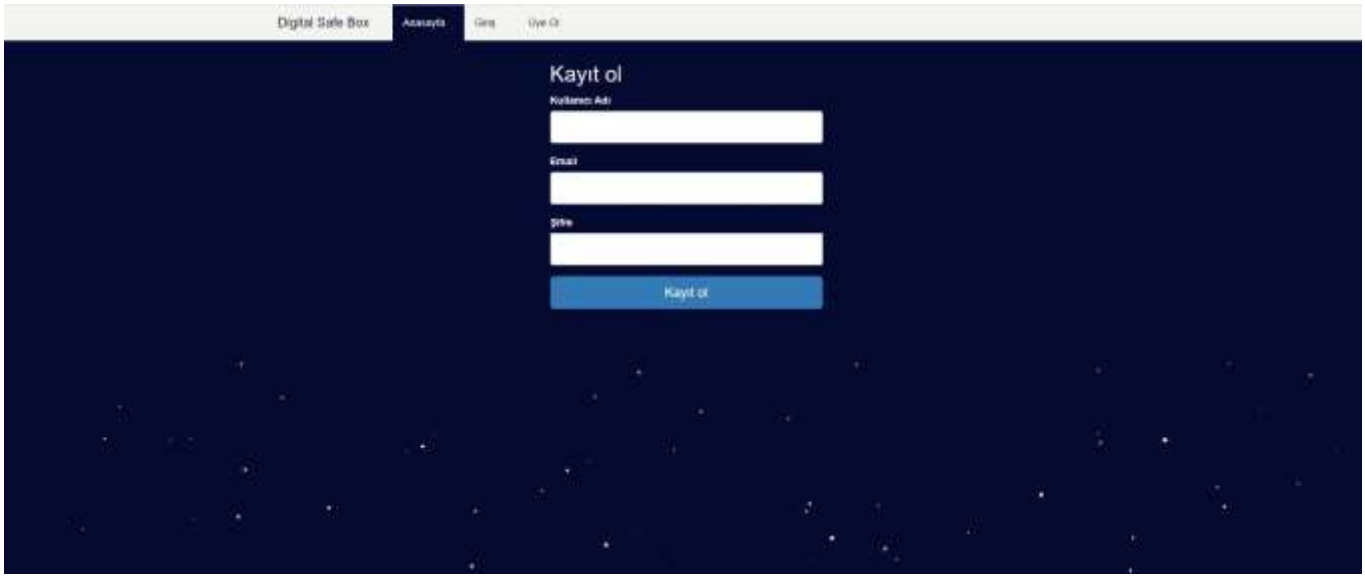
5. PROJE WEB SİTESİNİN KULLANIMI

Proje web site linkimize bağlanıldığında Resim-1'de görülen Anasayfa ekranı kullanıcıyı karşılamaktadır.



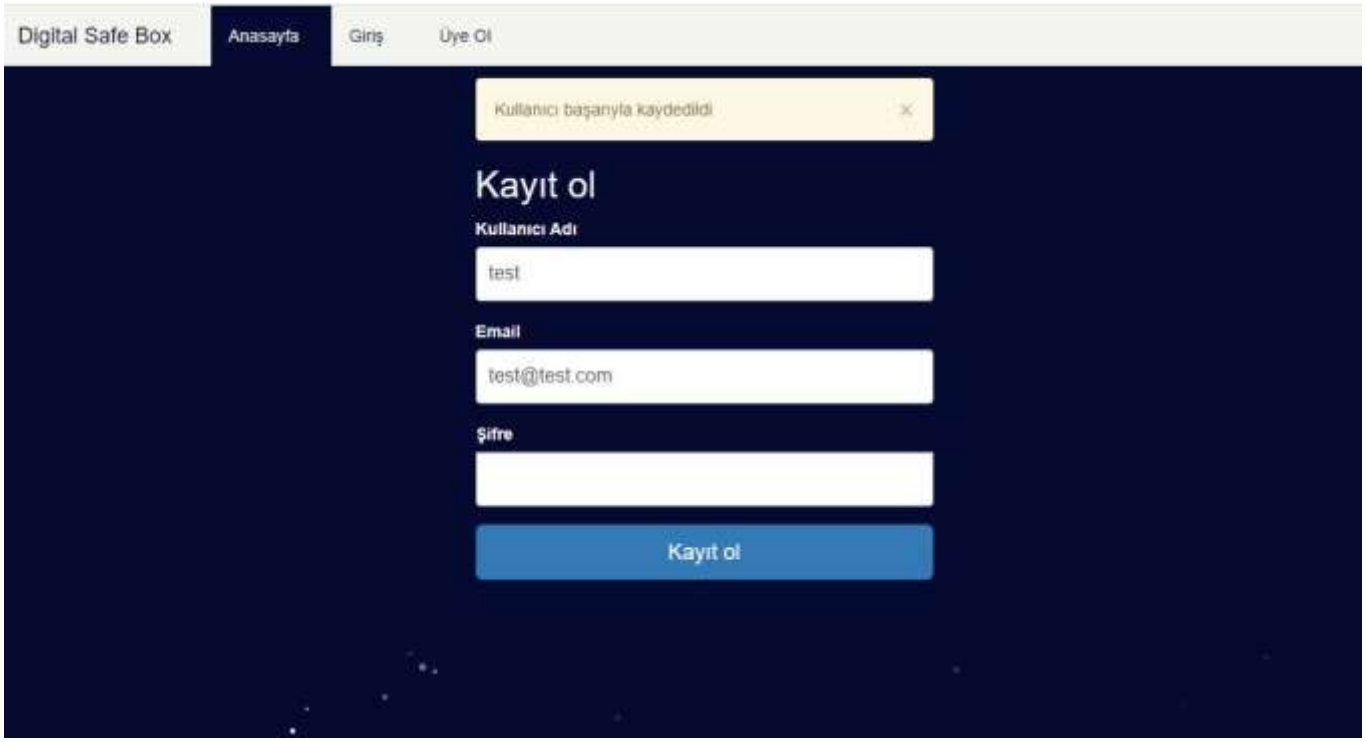
Resim 1. Proje anasayfası

Kullanıcın sitenin içeriğinden faydalanması için siteye üye olması gerekmektedir. Üye olma alanı Anasayfada bulunan "Üye Ol" linki üzerinden gerçekleşmektedir. Kullanıcılar Resim-2 'de gösterilen kullanıcı adı, email ve şifre alanlarını doldurarak siteye üye olabilmektedir.

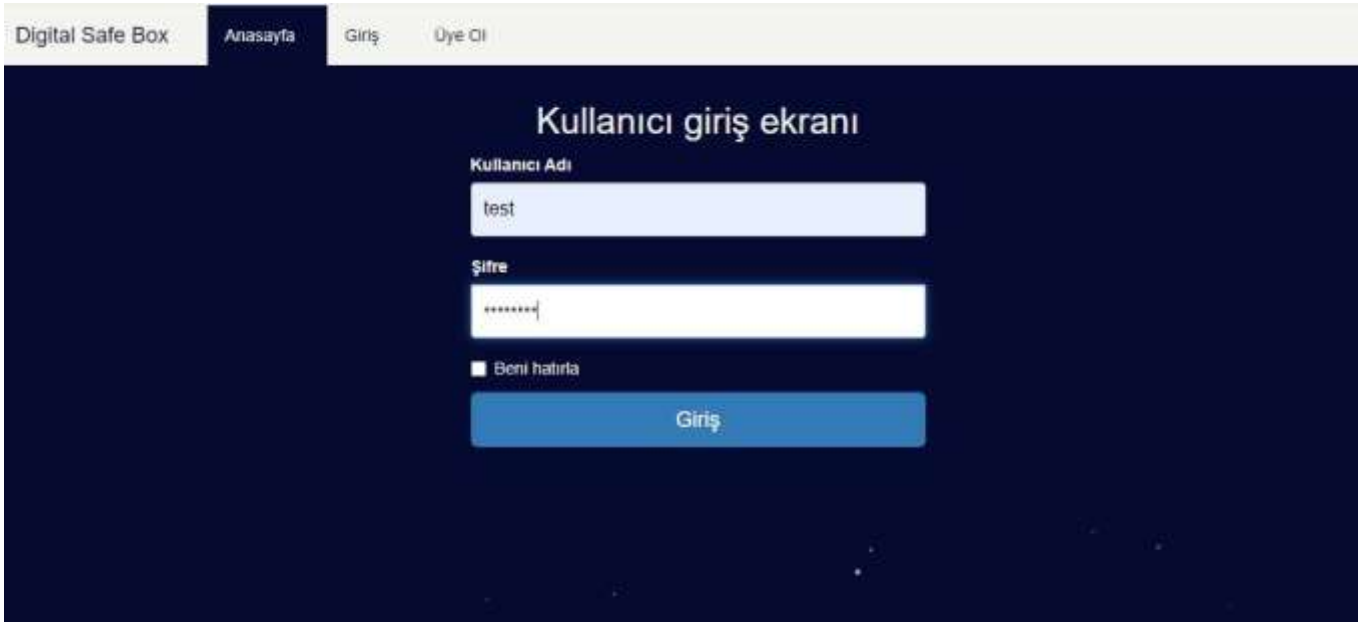


Resim 2. Site kayıt ol sayfası

Siteye kayıt olan kullanıcıların başarı ile kayıt olmaları durumunda ekran bildirimi kendilerine iletilmektedir. Başarılı şekilde kayıt olan kullanıcılar Navigasyon çubuğu üzerinde bulunan 'Giriş' alanından kullanıcı adı ve şifre bilgilerini girerek giriş yapabilmektedir. (Resim 3 ve Resim 4)



Resim 3. Kayıt olma işlemi



Resim 4. Kullanıcı giriş ekranı

Kullanıcılar siteye giriş yaptıktan sonra kendilerini dashboard alanı karşılamaktadır. Bu alan kullanıcıya aittir ve diğer kullanıcılar tarafından görülememektedir. Kullanıcılar dosya yükleme alanından verilerini Web site veritabanına kendi belirledikleri şifreler ile kaydedebilmektedir. Veritabanına eklenen belgeler (fotoğraf, doküman vb.) Resim-5'te sayfa alt kısmında görülen Dosya bilgileri kısmında görülebilmektedir. Kullanıcılar dashboardda bulunan alandan dosya yükleme alanına istedikleri dosyayı sürükleyebilir veya alana tıklayarak Resim-6'da görülen dosya yükleme penceresinden istedikleri dosyayı siteye yükleyebilmektedirler.

Resim 5. Dashboard alanı

Resim 6. Dosya yükleme alanı kullanımı

Resim-7 ile web sitemize örnek-resim.jpg isimli resim dosyamız yüklenmiştir. Dosya yükleme işleminden vazgeçtiğimizde Remove! butonuna tıklanabilmektedir. Başka bir dosya yüklemek için dosya yükleme işlemine tekrar edilebilir. Dosya yükleme işleminden sonra alt kısımda bulunan metin alanına dosyamızı şifrelemek istediğimiz şifre bilgisi girilerek (Resim-7'de "deneme" olarak girilmiştir) ve Upload! butonuna tıklanarak dosyamızın şifreleme işlemi başarıyla gerçekleştirilmektedir. (Resim-8)

Resim 7. Dosya yükleme alanı kullanımı

Resim 8. Upload işlemi sonrası ekran bildirimi

Kullanıcılar dosyalarını şifreledikten sonra şifrelenen dosyalar Resim-9'da beliritlen Dashboard alanında dosya ismi, dosya işlem tarihi ve ID numalarıyla görülebilmektedir. (Burada görülen key bilgisi bilgi amacıyla eklenmiştir ve Key bilgisi kullanıcıya mail yoluyla iletilecektir)

Resim 9. Şifrelenen dosyaların Dashboard alanında görüntülenmesi

Kullanıcılar şifreledikleri dosyayı tekrar bilgisayarlarına indirmek istedikleri takdirde, dosya ID bilgisi ve dosya şifre bilgisini Resim-10'da belirtilen ilgili alanlara girerek ve download butonuna tıklanılarak dosyanın şifre çözümleme ve bilgisayara indirme işlemlerini gerçekleştirebilmektedir.

Resim 10. Şifrelenen dosyaların indirilmesi işlemi

Dosya şifre çözüm işlemi sonrası bilgisayarımıza “deneme” isminde indirilmektedir (Resim-11) ve indirilen dosyanın uzantısı manuel olarak (.jpg .txt vb.) düzeltildikten sonra dosyamız ilk halini almaktadır.

Resim 11. Deneme isimli dosyanın bilgisayara indirilmesi

Web sitemize yüklenen dosyaların ve şifre bilgisinin şifrelemesiyle ilgili işlemlerin kaynak kod kısmı Resim-12’de ayrıntılı şekilde belirtilmiştir. Kullanıcıdan request edilen file isimli dosya ve kullanıcıdan alınan key bilgisi userfilepass ile Python app dosyamıza iletilmektedir. Sonrasında Fernet ile oluşturulan key ile dosyamız şifrelenmektedir ve kullanıcıdan aldığımız userfilepass şifre bilgisi ile Fernet ile şifrelenen dosyamız AES şifremele algoritması ile tekrar şifrelenerek newFile adı altında veritabanımızda bulunan FileContents alanına kayıt edilmektedir.

```
def upload():

    if request.method == 'GET':
        pass
    else:
        userfilepass = request.form['userkey']

        for file in request.files.getlist("file"):
            file=request.files["file"]

        if file.filename == '':
            flash('Lütfen bir dosya yükleyiniz!')
            return render_template("dashboard.html")

        else:
            target=app.config["SAFEBOX"]
            uploadfile=app.config["FILE_UPLOAD"]



            key = Fernet.generate_key()
            f = Fernet(key)


            file.save(os.path.join(app.config["FILE_UPLOAD"],file.filename))


            with open(app.config["FILE_UPLOAD"]+"\\"+file.filename,"rb") as new_enc_file:
                original=new_enc_file.read()
                original2 = f.encrypt(original)
                original3=str(original2)


            encrypted=AESCipher(userfilepass).encrypt(original3).decode('utf-8')


            newFile=FileContents(name=file.filename, data=original, edata=encrypted, key=key, user=current_user.username, date=datetime.datetime.now())
            db.session.add(newFile)
            db.session.commit()


            myfiles=FileContents.query.all()
```

Resim 12. Dosya ve kullanıcı şifresinin şifreleme işlemi kaynak dosyası

6. PROJE SUNUMU, KAYNAK DOSYALARI ve WEB LİNKİ

Sunum : https://youtu.be/dR2duxocB1Y
Kaynak Dosyalar : https://github.com/xsmileh/ymgk_digitalsafebox
Web linki : https://ymgkdijitalsafebox1.herokuapp.com/

7. KAYNAKLAR

[1]. https://tr.wikipedia.org/wiki/Python
[2]. https://tr.wikipedia.org/wiki/HTML
[3]. https://tr.wikipedia.org/wiki/CSS
[4]. https://tr.wikipedia.org/wiki/JavaScript
[5]. https://tr.wikipedia.org/wiki/Sublime_Text
[6]. https://tr.wikipedia.org/wiki/Bootstrap_(önyüz_çatısı)
[7]. https://en.wikipedia.org/wiki/Flask_(web_framework)
[8]. https://cryptography.io/en/latest/fernet/
[9]. https://tr.wikipedia.org/wiki/AES
[10]. https://en.wikipedia.org/wiki/SQLAlchemy