T.C.

FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ

YMH 459 – YAZILIM MÜHENDİSLİĞİNDE GÜNCEL KONULAR

GÖRÜNTÜ ŞİFRELEME UYGULAMASI

GÜNDÜZ GRUP 1

OCAK 2021

İçindekiler

1. Giriş

1.1. Proje Amacı

Bu çalışma ile kullanıcının her platformdan erişebileceği ve seçtiği bir görüntünün güvenli ve hızlı bir şekilde şifrelenmesi ve daha sonrasında istediği herhangi bir zamanda şifresinin çözülerek orijinal görüntüyü elde edebileceği bir platform amaçlanmaktadır.

1.2. Proje Kapsamı

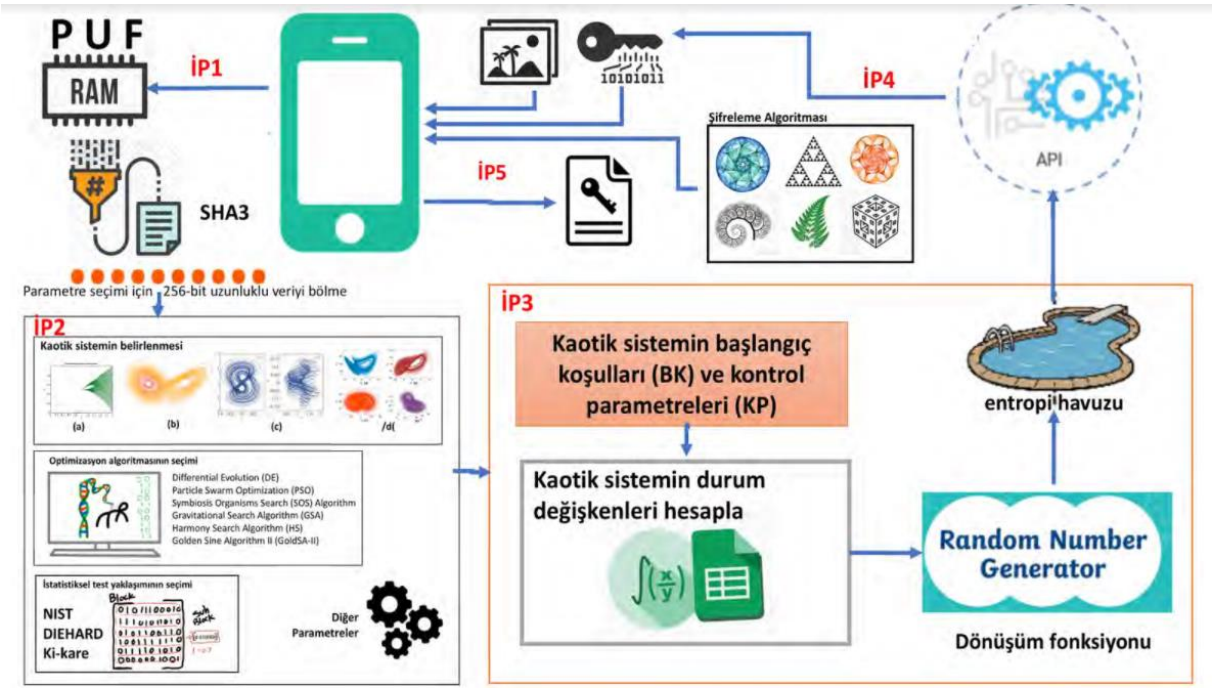
Bahsi geçen platformda en önemli kriter güvenlik olurken bununla birlikte bir o kadar önemli noktalar da bu proje kapsamında ele alınmıştır. Bu noktalar erişilebilirlik yani mobil veya masaüstü cihazlardan platformu kullanabilmek, kullanıcı dostu yani son kullanıcının hiçbir teknik bilgiye gereksinim duymadan platformu kullanabilmesi ve hız olarak karşımıza çıkmaktadır. Proje kapsamında Doç. Dr. Fatih Özkaynak sorumluluğunda gerçekleştirilen “An Image Encryption Algorithm Based on Chaotic Selection of Robust Cryptographic Primitives” adlı akademik çalışma baz alınmıştır. Gerçekleştirilen çalışma bahsi geçen akademik çalışmanın tamamen bir uygulaması olmayıp, tarafımızca majör ve minör değişiklikler uygulanmıştır.

2. Proje Planı

2.1. Giriş

Görüntü şifreleme, bir görüntünün piksellerinin değerlerinin veya pozisyonlarının değiştirilmesi ile ortaya karmaşık bir görüntü çıkarılması işlemidir. Bu işlem gizlilik, mahremiyet, güvenlik ve daha bir çok amaç için gerçekleştirilmektedir.

2.2. Planlama



Fotoğraf 1: Önerilen Sistem Mimarisi

Önerilen sistem mimarisi incelendiğinde dağıtık bir sistem göze çarpmaktadır. Mobil cihaz aracılığıyla bir PUF üretilirken web platformunda rastgele sayı üretme işlemi ve aradaki iletişimi sağlayabilmek adına ise bir API görülmektedir. Bu proje kapsamında bu mimarinin dışına çıkılarak aynı gereksinim ve işlevlerin daha iyi bir şekilde gerçekleştirilmesi planlanmaktadır. Bu doğrultuda proje tamamen web platformunda gerçekleştirilecektir.

2.3. Zaman-İş Planı

İŞ/ZAMAN	1.HAFTA	2.HAFTA	3.HAFTA	4.HAFTA	5.HAFTA	6.HAFTA	7.HAFTA	8.HAFTA
PLANLAMA								
SİSTEM ANALİZİ								
MANTIKSAL SİSTEM OLUŞTURULMASI								
GERÇEKLEŞTİRİM								
DOĞRULAMA								

Fotoğraf 2: Zaman İş Planı

2.4. Proje Ekip Yapısı

DEVELOPER	ARAŞTIRMA	FRAKTAL	BELGELEME	TASARIM VE MOBİL
Recep Tayyib Aksakal	Nihat Çetin	Mertbaba Okulmuş	Abdullah Sait Koç	Pınar Savcı
Melahat Erbaş	Mehmet İpek	Yekbun Demir	Metehan Meteoğlu	Selin Aslanbulut
Tuğşad Öcal	Abdullah Binyad Soysal	Şeyma Hocaoğlu	Cemil Ahlatcı	Zeynel Abidin Sarısaltık
		Muhammed Tenlik		Mehmet Kart

Tablo 1: Proje Ekip Yapısı

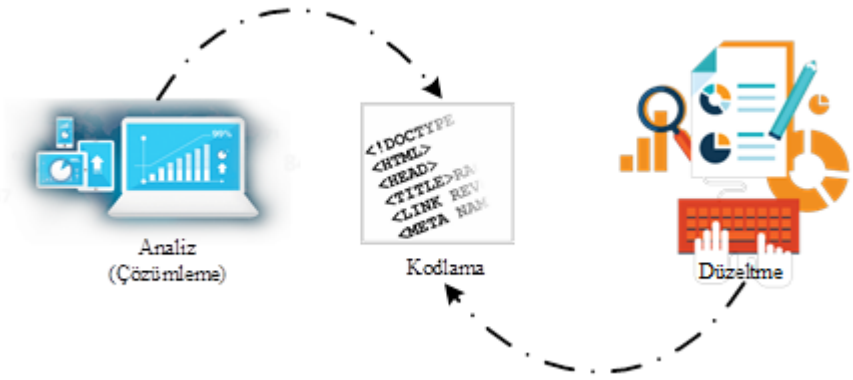
2.5. Kullanılan Özel Geliştirme Araçları ve Ortamları

KULLANILAN ARAÇ	KULLANIM AMACI
VİSUAL STUDİO CODE	IDE
PYTHON V3	Programlama Dili
PİLLOW	Görüntü İşlemleri
NUMPY	Matris Hesaplamaları
FLASK	Web Micro Framework
BOOTSTRAP	Arayüz Tasarımı
AJAX	Client-Backend Haberleşmesi
JQUERY	DOM İşlemleri

Tablo 2: Kullanılan Araçlar ve Ortamlar

2.6. Yöntem ve Metodolojiler

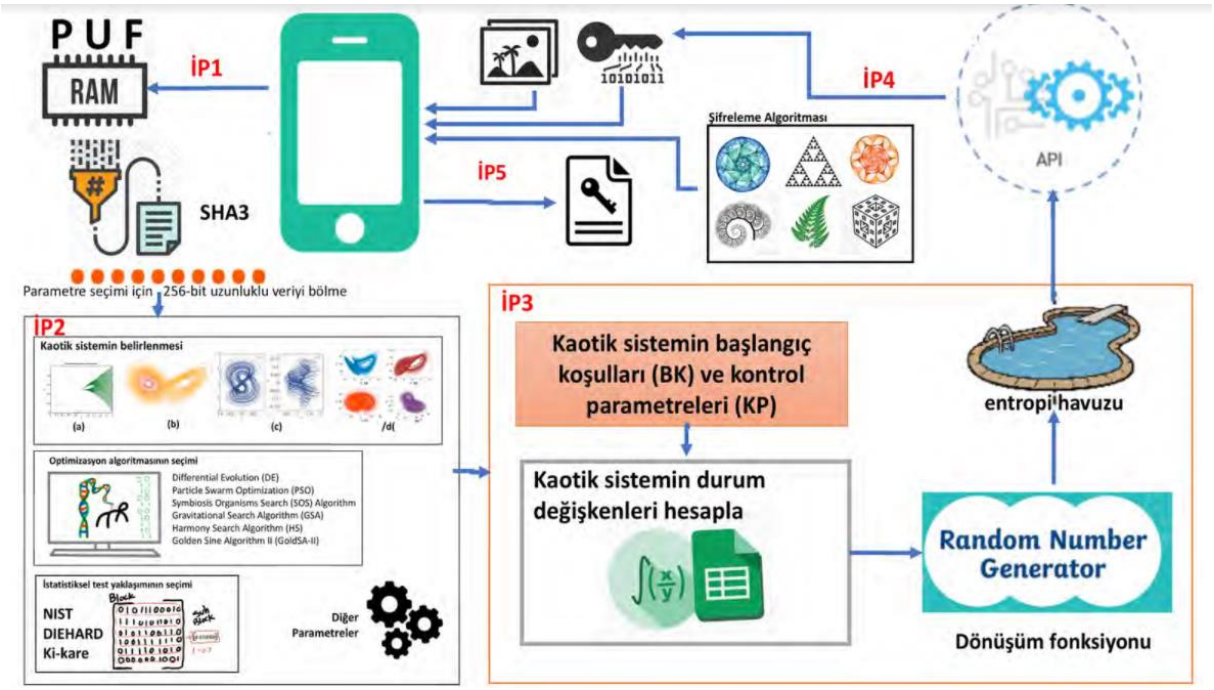
Proje kapsamında geliştirme yöntemi olarak Kodla ve Düzelt benimsenmiştir. Kodla ve Düzelt yöntemi 3 aşamadan oluşmaktadır. Bu aşamalar; analiz, kodlama ve düzeltme şeklindedir. Proje gereksinimleri keskin bir biçimde belirliyse ve çok fazla revizyon veya değişiklik ön görülmüyorsa hızlı geliştirme ve ortaya somut bir uygulama çıkması açısından oldukça yararlıdır.



Fotoğraf 3: Kodla ve Düzelt

3. Sistem Çözümleme

3.1. Mevcut Sistem İncelemesi

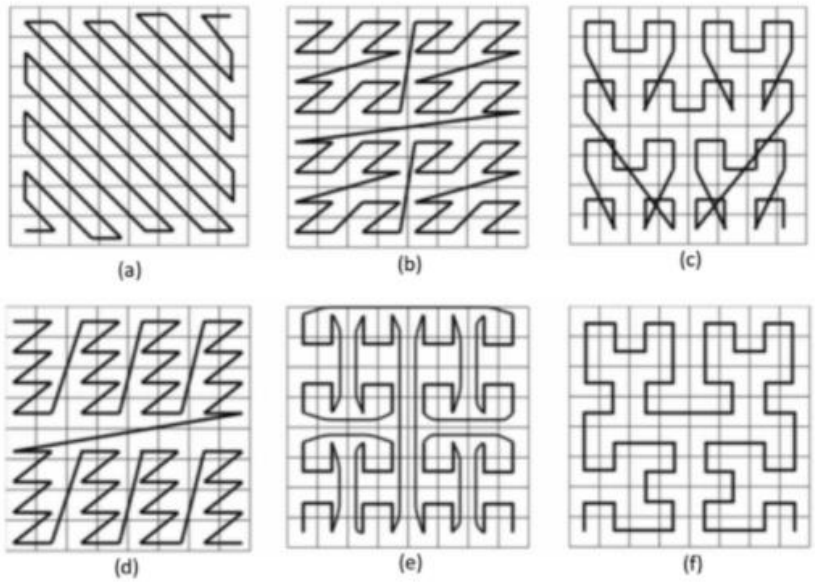


Fotoğraf 4: Mevcut Önerilen Sistem Mimarisi

Mevcutta önerilen sistem incelendiği zaman çeşitli iş paketlerine ayrılmış multi platform bir mimari görünmektedir. Önerilen sistemin kilit noktalarına baktığımızda ise RAM kaynaklı bir PUF üretilmesi beklenmekte ve rastgele bir üretilmesi işlemleri göze çarpmaktadır. Bununla birlikte fraktal desenlerinin kullanımı ve çeşitli iletişim katmanları da yer almaktadır.

Önerilen sistemde fraktal kullanılmasının amacını inceleyecek olursak, bu yöntem aslında güvenliği artırabilmek amacıyla kullanılmaktadır. Bazı görüntülerde şifreleme yapıldığında yine de o görüntü görünebiliyor ve tam olarak şifreleme sağlanamıyor.

Şekil 5. Sayısal görüntülerin şifrelenmesi sürecindeki korelasyon problemi



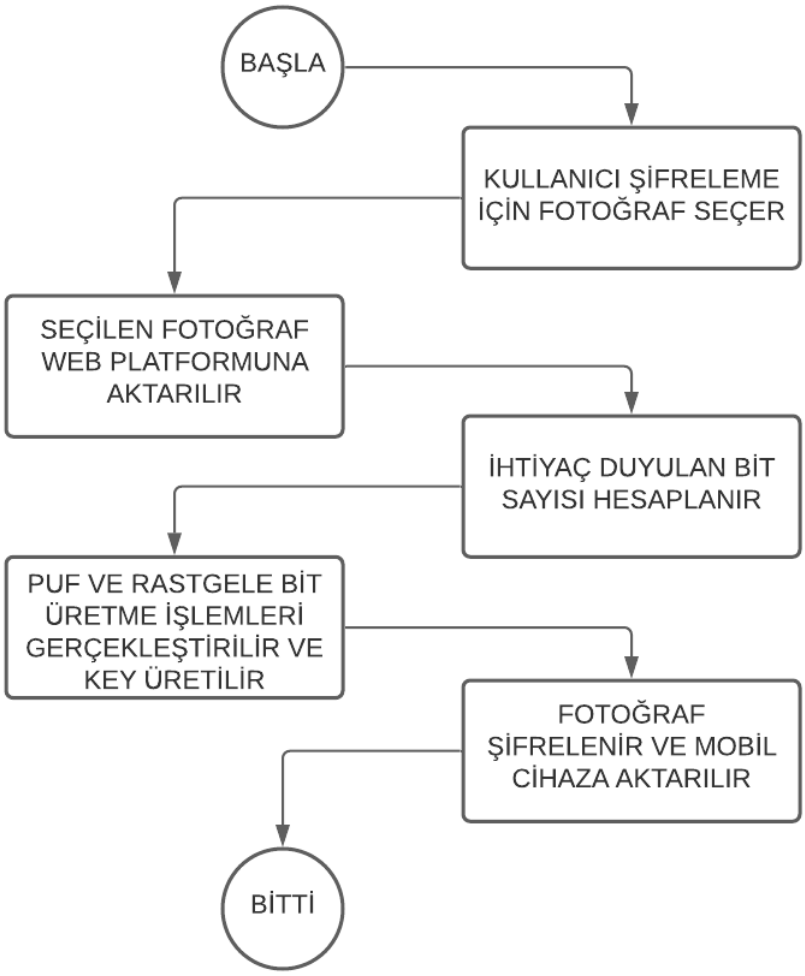
Şekil 6. Örnek fraktal desenler

Fotoğraf 5: Örnek Fraktal Desenleri ve Kullanım Amacı

Fotoğraf 4'de de görülebildiği üzere üzerinde şifreleme gerçekleştirilen görüntüler yine de belli olmakta ve ortaya bir kolerasyon problemi çıkmaktadır. Görüntüler aslında iki boyutlu bir matris olduğundan ve bu matrisi okurken çeşitli desenler kullanarak okursak bu olumsuz durumu ortadan kaldırabiliriz amacıyla fraktal desenleri görüntü şifreleme de kullanılmaktadır.

3.2. Önerilen Sistem İncelemesi

Önerilen sistem incelenmiş ve kritik noktalar belirlenmiştir. Sergilenecek olan yaklaşımda PUF verisinin üretilme işleminin web platformunda gerçekleştirilmesi hız, sistem bütünlüğü ve güvenli veri yolu gibi kıstaslar açısından daha avantajlı olabileceği düşünülerek web platformuna taşınmıştır. Hedeflenen sistemde bütün işlemlerin web platformundan gerçekleştirilmesi ve mobil uygulama aracılığı ile sadece kullanıcının fotoğraf seçme işlemi gerçekleştirmesi sonrasında ise bu fotoğrafın web platformuna aktarılarak PUF ve rastgele bit üreterek key verisinin elde edilmesiyle şifrelenmesi ve sonrasında ise mobil cihaza geri aktarılması planlanmıştır.

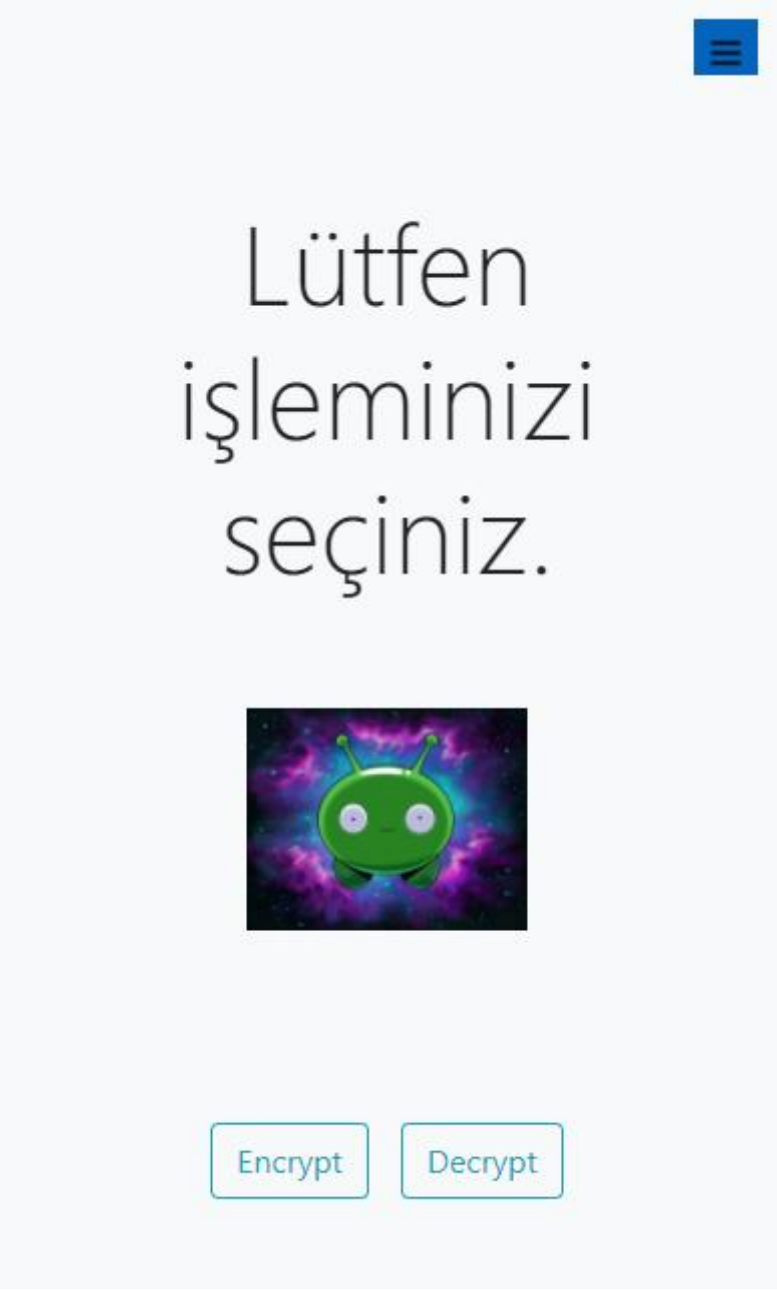


Fotoğraf 6: Sistem Akış Diyagramı

3.2.1. Güvenli Veri Yolu

Güvenli veri yolu sorunu SSL kullanılması halinde ortadan kalkmış olmasıyla beraber PUF ve rastgele bit üretme işleminin görüntüdeki her bit için üretilerek mobil cihaza oranla daha hızlı ve daha güvenli bir sonuç ortaya çıkarılması sağlanacaktır.

3.2.2. Sistem Arayüzü



Fotoğraf 7: Sistem Arayüzü

Fotoğraf 8: Şifrelenmiş Görüntü

Fotoğraf 9:Şifresi Çözülen Görüntü

3.2.3. Decrypt İşlemi

Decrypt işleminin gerçekleştirilebilmesi açısından üretilen keylerin saklanması gerekmektedir. Şifrelemeyi üretilen key ile görüntüyü XORlayarak gerçekleştirdiğimiz için şifrelenmiş fotoğrafı tekrar XORlayarak orijinal görüntüyü elde edebiliriz. Ancak bunun için öncelikle keylerin saklanması ve ardından bir şifrelenmiş görüntü decrypt edilmek istendiğinde hangi key ile decrypt edileceğini tespit etmek gerekmektedir.

Şifrelenmiş görüntü ile key ilişkilendirmesini görüntü şifrelendikten hemen sonra md5 kullanarak şifrelenmiş görüntünün hashini alarak sağlanmalıdır. Keyi saklamak için veritabanı kullanılabilir fakat büyük görüntülerde performans düşüşü yaşanacağı için bu yöntem pek uygun değildir. Üretilen key aslında sisteme şifrelenmesi için yüklenen görüntü ölçülerinde rastgele üretilmiş bir görüntüdür. Bu bağlamda üretilen keyi bir fotoğraf olarak dizinde saklamak en doğru seçim olacaktır.

4. Sistem Tasarımı

4.1. Sistem Mimarisi

4.1.1. Mobil Uygulamadan Vazgeçilmesi

Bütün işlevlerin gerek performans gerek geliştirme kolaylığı gerekse güvenlik kaynaklı olarak web platformuna taşınması ve mobil cihaz tarafında sadece fotoğraf alışveriş işlevi kalması dolayısıyla aynı işlem web platformunda responsive bir tasarım gerçekleştirerek sağlanabilir. Bundan dolayı mobil cihazlar için geliştirme yapmaktan vazgeçilmiştir. Bu durumda bütün bu projeyi web platformuna taşıyarak birçok avantaj kazanılmıştır. Bu avantajlar şu şekildedir:

- PUF verisi üretmek daha hızlı ve görüntüdeki her bit için bit üretilebilmektedir.
- Rastgele bit ve PUF verileri aynı platformda yer almaktadır böylelikle böylesine büyük dosyaların transferi sorun olmaktan çıkmıştır.(4k bir fotoğraf için 25 mb)
- SSL kullanılması durumunda güvenli veri yolu kolaylıkla sağlanabilmektedir.
- Kullanıcı görüntü şifrelemek için illa cihazına bir uygulama kurmak zorunda değildir.
- Responsive tasarım ile bütün cihazlardan sisteme erişilebilmektedir.
- Kullanıcı tarafında herhangi bir bilgi bulunmadığından bütün bilgiler web platformunda yer aldığından ayrıcalıklı saldırgan tehdidi azaltılmıştır.

4.1.2. Fraktal Desenlere İhtiyaç Duyulmaması

Hem PUF verisi üretilirken hem de rastgele bit üretilirken görüntüdeki her bit için bir bit üretildiğinden dolayı fraktal kullanma ihtiyacı duyulmamıştır.

Fotoğraf 10: Fraktallara İhtiyaç Duyulmaması

Fotoğraf 11: Fraktallara İhtiyaç Duyulmaması

Fotoğraf 12: Fraktallara İhtiyaç Duyulmaması

4.1.3. Web Platformu

Web platformunda bir Python micro frameworkü olan Flask kullanılacaktır. Flask ihtiyaçlarınız doğrultusunda ek paketler ile genişletebileceğiniz ve gereksiz özellikler veya kütüphaneler barındırmayan çıplak bir frameworktür. Geliştirilecek olan sistem kapsamında herhangi bir veritabanı ihtiyacı bulunmamakla beraber kullanıcıya geri bildirim verebilmek ve client backend arasındaki iletişimi sağlayabilmek adına ise AJAX kullanılacaktır.

Fotoğraf 13: Web Platformu

5. Sistem Gerçekleştirimi

5.1. Yazılım Geliştirme Ortamı

Proje kapsamında yazılım geliştirirken IDE olarak VS Code kullanılmıştır. Çeşitli tarayıcılar ile de çıktılar kontrol edilmiştir. Proje kapsamında tercih edilen dil Python olurken web geliştirme işlemini gerçekleştirebilmek adına ise Flask micro frameworkü kullanılmıştır. Bununla birlikte çeşitli Python kütüphaneleri de proje kapsamında kullanılmıştır. Geliştirme ortamı verilen tabloda görülebilmektedir.

Kullanılan Araç	Kullanım Amacı
Visual Studio Code	IDE
Python v3	Programlama Dili
Pillow	Görüntü İşlemleri
Numpy	Matris Hesaplamaları
Flask	Web Micro Framework
Bootstrap	Arayüz Tasarımı
AJAX	Client-Backend Haberleşmesi
Jquery	DOM İşlemleri

Tablo 3: Yazılım Geliştirme Ortamı

Proje kapsamında herhangi bir veritabanı ihtiyacı doğmadığından dolayı veritabanı kullanılmamıştır.

5.2. Sunucu

Gerçekleştirilen sistem Ubuntu 20.04 üzerinde Nginx ve Gunicorn aracılığıyla ayağa kaldırılmıştır. Bu süreçte digitalocean tarafından yayınlanan "How to Serve Flask Applications with Gunicorn and Nginx on Ubuntu 20.04" başlıklı makalesi takip edilmiştir.

Bu bağlamda Google Cloud üzerinde bir sanal sunucu oluşturularak Ubuntu üzerinde kurulum şu şekilde gerçekleştirilmiştir.

```
$ sudo apt update
$ sudo apt install python3-pip python3-dev build-essential libssl-dev libffi-dev python3-setupt
```

```
$ sudo apt install python3-venv
```

Fotoğraf 14, 15: Gerekli Paketlerin Kurulumu

```
$ source myprojectenv/bin/activate
```

Fotoğraf 16: Virtual Environmentin Aktive Edilmesi

```
$ pip install wheel
```

```
(myprojectenv) $ pip install gunicorn flask
```

Fotoğraf 17,18: Gerekli Python Kütüphanelerinin Kurulması

```
from myproject import app

if __name__ == "__main__":
    app.run()
```

Fotoğraf 19: Python WSGI dosyasının oluşturulması

```
[Unit]
Description=Gunicorn instance to serve myproject
After=network.target

[Service]
User=sammy
Group=www-data
WorkingDirectory=/home/sammy/myproject
Environment="PATH=/home/sammy/myproject/myprojectenv/bin"
ExecStart=/home/sammy/myproject/myprojectenv/bin/gunicorn --workers 3 --bind unix:myproject.soc

[Install]
WantedBy=multi-user.target
```

Fotoğraf 20: Service dosyasının oluşturulması

```
server {
    listen 80;
    server_name your_domain www.your_domain;

    location / {
        include proxy_params;
        proxy_pass http://unix:/home/sammy/myproject/myproject.sock;
    }
}
```

Fotoğraf 21: Nginx Server Dosyasının Oluşturulması

6. Doğrulama ve Geçerleme

6.1. Sınama Yöntemleri

Gerçekleştirilen sistemde gerek geliştirme aşamasında gerekse geliştirme tamamlandıktan sonra çeşitli testler gerçekleştirilmiştir. Bu testlerde çeşitli çözünürlükteki fotoğraflar kullanılmıştır. 1024x805 bir fotoğraf için 19 saniye şifreleme 6 saniye şifre çözme gibi performans sonuçları elde ederken 4k bir fotoğraf için ise 3 dakika şifreleme 1 dakika şifre çözme gibi performans sonuçlar elde ettik.

Bununla birlikte kolerasyon problemi doğurabilecek fotoğraflar ile denemeler yapıldığında ise fraktal desenler kullanılmamasına rağmen herhangi bir kolerasyon problemi görülmemiştir.

Gerçekleştirilen sistem beklenen isterleri tam bir şekilde sağlamaktadır.