

File Backup System

I. Giriş

Bu proje python kullanılarak dosya yedekleme ve depolama sistemi oluşturulmuştur. Gerekli yerlerde process ve thread işlemleri uygulanmıştır.

Sistemde 2 adet rol vardır. Bu roller “Sistem Yöneticisi” ve “Bireysel Kullanıcı” şeklindedir.

Sistem Yöneticisi, kullanıcının dosyalarını ve paylaştığı dosyaları ve dosyaları kimle paylaştığını görebilir. Ayrıca Sistem Yöneticisi bireysel kullanıcı tarafından gönderilen şifre değiştirme taleplerini görebilir, onaylayabilir veya reddedebilir. Sistem yöneticisi belirlenen anormal durumları gözlemleyebilir. Sistem Yöneticisi kullanıcının yükleyebileceği toplam dosya boyutunu belirleyebilir, bir kullanıcıyı silebilir, kullanıcının usernamini ve şifresinin hashlenmiş halini görebilir.

Bireysel Kullanıcılar sisteme kayıt olabilir, giriş yapabilir, sisteme dosya yükleyebilir, kendisi ile ilişkili üye ekleyebilir, ilişkili üyelere dosya gönderebilir ve ilişkili üyelerle dosya düzenleyebilir, şifre değiştirme talebinde bulunabilir, ve usernamini değiştirebilir. Kullanıcının yüklemiş olduğu dosyalar otomatik olarak yedeklenmektedir.

Proje Amacı

Bu proje, kullanıcıların dijital dosyalarını güvenli ve düzenli bir şekilde yedekleyip senkronize etmelerine olanak sağlayan bir Dosya Depolama-Yedekleme Sistemi geliştirmeyi amaçlamaktadır. Aynı zamanda sistem, kullanıcı davranışlarını analiz ederek güvenlik ihlallerini tespit etmeyi, gerçek zamanlı log izleme ve anormallik tespit mekanizmaları ile olası tehditleri hızlıca tespit edip önlem almayı hedeflemektedir. Projenin odak noktası; kullanıcı deneyimini iyileştirmek, verilerin bütünlüğünü korumak ve olası güvenlik açıklarını minimize etmektir.

II. Yöntem

A) Veri Tabanı Yapısı

Bu projede veritabanı olarak sqllite kullanılmıştır. Projede toplam 8 adet tablo bulunmaktadır. Bu tablolar şu şekildedir:

1) User Tablosu:

a) **Amaç:** Kullanıcı bilgilerini güvenli bir şekilde saklamak.

b) Alanlar:

- Id: Her kullanıcı için benzersiz bir tanımlayıcı.
- username: Kullanıcı adı (eşsiz ve boş olamaz).
- password_hash: Şifrelerin güvenli bir şekilde saklanması için kullanılan hash değeri.
- salt: Hash işlemi için kullanılan rastgele veri.
- role: Kullanıcının rolünü belirtir (admin veya bireyselkullanıcı).

2) Password_Reset_Requests Tablosu

a) **Amaç:** Şifre sıfırlama taleplerini takip etmek.

b) Alanlar:

- id : Her talep için benzersiz bir tanımlayıcı.
- user_id: Talepte bulunan kullanıcıyı işaret eder (users tablosundaki id alanına referans).
- status: Talebin durumunu belirtir (pending, approved, rejected).

3) User_Files Tablosu

a) **Amaç:** Kullanıcıların yüklediği dosyaları kaydetmek.

b) Alanlar:

- id: Her dosya için benzersiz bir tanımlayıcı.
- user_id: Dosyayı yükleyen kullanıcıyı işaret eder (users tablosundaki id alanına referans).
- file_name: Dosyanın adı.
- file_path : Dosyanın sistemdeki tam yolu.

4) File_Backups Tablosu

a) Amaç: Kullanıcı dosyalarının yedeklerini kaydetmek.

b) Alanlar:

- **id:** Her yedekleme kaydı için benzersiz bir tanımlayıcı.
- **file_id:** Yedeklenen dosyayı işaret eder (user_files tablosundaki id alanına referans).
- **backup_path:** Yedekleme dosyasının saklandığı yol.
- **log_path:** Yedekleme işlemiyle ilgili log dosyasının yolu.
- **timestamp:** Yedekleme işleminin gerçekleştiği tarih ve saat.

5) User_Relationships Tablosu

a) Amaç: Kullanıcılar arasında ilişkileri (örneğin dosya paylaşımı için) modellemek.

b) Alanlar:

- **id:** Her ilişki için benzersiz bir tanımlayıcı.
- **user_id:** İlişkiyi başlatan kullanıcı (users tablosundaki id alanına referans).
- **related_user_id:** İlişkideki diğer kullanıcı (users tablosundaki id alanına referans).
- **UNIQUE:** Aynı ilişkiyi tekrar kaydetmeyi önler.

6) File_Shares Tablosu

a) Amaç: Kullanıcılar arasındaki dosya paylaşım işlemlerini kaydetmek.

b) Alanlar:

- **id:** Her paylaşım için benzersiz bir tanımlayıcı.
- **sender_id:** Dosyayı paylaşan kullanıcı (users tablosundaki id alanına referans).
- **receiver_id:** Dosyayı alan kullanıcı (users tablosundaki id alanına referans).
- **file_name:** Paylaşılan dosyanın adı.
- **file_path:** Dosyanın sistemdeki tam yolu.
- **timestamp:** Paylaşımın gerçekleştiği tarih ve saat.
- **modified_name:** Dosyanın kullanıcı tarafından yeniden adlandırılmış hali.

7) Notifications Tablosu

a) Amaç: Kullanıcılara yönelik bildirimleri saklamak.

b) Alanlar:

- **id:** Her bildirim için benzersiz bir tanımlayıcı.

- **user_id:** Bildirim alan kullanıcı (users tablosundaki id alanına referans).
- **message:** Bildirim içeriği.
- **seen:** Bildirimin görülüp görülmediği (0: Görülmedi, 1: Görüldü).
- **Timestamp:** Bildirimin oluşturulduğu tarih ve saat.

8) Admin_Notifications Tablosu

a) Amaç: Yöneticiye özel bildirimleri kaydetmek.

b) Alanlar:

- **id:** Her bildirim için benzersiz bir tanımlayıcı.
- **message:** Bildirim içeriği.
- **status:** Bildirimin durumu (unread veya read).

B) Sistem Yöneticisi

Sistem yöneticisi arayüzü ve işlevleri oluşturulurken [1] deki Sistem yöneticisi için verilen istelere göre gerçekleştirilmiştir.

Sistem Yöneticisinin Görevleri

- Şifre talep izleme
- Kullanıcı depolama limitlerini control etme
- Kullanıcı profillerini inceleme
- Log dosyalarına erişim
- Anomaly Bildirimlerini alma

1- Şifre talep izleme: Sistem yöneticisi Kullanıcı tarafından gönderilen şifre taleplerini görebilir, onaylayabilir veya reddedebilir. Reddetmesi durumunda ilgili kullanıcıya talebin reddedildiği bildirimi gönderilir. Onaylaması durumunda ilgili kullanıcının arayüzünde “Set Password” butonu çıkar ve artık kullanıcı yeni şifresini oluşturabilir.

İlgili Fonksiyonlar

- a) **show_admin_dashbord():** Kullanıcı giriş yaptığında rolü admin ise bu fonksiyon çalışır ve admin arayüzü kullanıcıya gösterilir.

- b) **show_password_reset_request(s):** Bu fonksiyon ile kullanıcıların şifre talepleri admin arayüzünde gösterilmiştir.
- c) **get_unique_pending_requests():** Bu fonksiyon ile aynı kullanıcının art arda göndermiş olduğu şifre değişikliği talepleri filtrelenir ve tek bir şekilde gösterilir.
- d) **approve_request():** Bu fonksiyon ile kullanıcının göndermiş olduğu talepleri admin onaylar.
- e) **reject_request():** Bu fonksiyon ile kullanıcının göndermiş olduğu talepleri admin reddeder.

2- Kullanıcı Yönetimi: Bu kısımda sistem yöneticisi kullanıcıları listeler. Kullanıcıların Id ,Username ve Limit bilgileri ile birlikte listelenir. Her kullanıcının yanında review, Set Limit ve Delete butonları bulunur.

İlgili fonksiyonlar

- a) **manage_users():**Kullanıcıları listeler. Her bir kullanıcının yanında Delete Set Limit ve Rewiev butonları bulunur. Delete ile Kullanıcı silinir. Set Limit ile kullanıcının toplam depolama limiti belirlenir. Review ile kullanıcının bilgilerinin inceleyebileceği sayfaya yönlendirilme işlemi gerçekleştirilir.
- b) **delete_user():** Kullanıcıyı siler.
- c) **set_storage_limit():** Kullanıcının Limiti bu fonksiyon ile belirlenir. Veri tabanına ilgili kullanıcının limiti yazılır.
- d) **ask_limit():** bu fonksiyon ile admine limiti belirleyebileceği pencere açılır.

e) **rewiev_user():** Admin incelenmek istenen kullanıcı bu fonksiyon ile ilgili kullanıcının username bilgisi, şifresinin hashlanmış hali, sistemdeki dosyalarını inceleyebilmek için Files isimli bir buton ve ilgili kullanıcının paylaştığı dosyaları ve hangi kullanıcı ile paylaştığını inceleyebilir.

f) **show_user_files():** Admin “Files” adlı butonuna tıkladığında bu fonksiyon çalışır ve ilgili kullanıcının dosyaları listelenir. Her dosyanın yanında open butonu ile dosyalar formatına uygun açılır.

g) **show_shared_files():**Admin “Shared Files” isimli butona tıkladığında bu fonksiyon çalışır ve ilgili kullanıcının paylaştığı dosyalar ve paylaşılan kişiler listelenir. Her dosyanın yanında open butonu ile dosyalar formatına uygun açılır.

3- Anormal Durum Bildirimleri: Sistem yöneticisi, sistemde bulunan anormallikleri bildirim şeklinde haberdar edilir.

İlgili Fonksiyonlar:

- a) **show_admin_notifications():** Admin arayüzdeki “Notifications” isimli butona bastığında bu fonksiyon tetiklenir ve anormal durum bildirimleri sistem yöneticisine gösterilir.Sistem yöneticisi open butonu ile, tarihi ile birlikte anormal durumları inceleyebilir.
- b) **check_anomaly_log():** Anomaly log dosyasını kontrol eder ve yeni kayıtları veritabanındaki “admin_notifications” tablosuna ekler.
- c) **add_notification():** Kontrol edilen anormal durum mesajı veritabanına ekleyen fonksiyon.

4- Log Dosyasının İncelenmesi: Sistem yöneticisi sistemdeki anormal durumların akıttı edildiği anomaly_log dosyasını inceleyebilir.

a) **show_log_files():** “Log Files” butonuna basıldığında bu fonksiyon tetiklenir. Log

dosyası listelenir ve open ile anomalylog dosyası açılır.

C) Bireysel Kullanıcı

Bireysel kullanıcının arayüzü ve işlevleri oluştururken [1] deki kriterler göz önünde bulundurularak oluşturulmuştur.

Bireysel Kullanıcının işlevleri:

- Şifre değişim talebinde bulunma
- Kullanıcı adı değiştirme
- Sisteme dosya yükleme
- Üye ekleme
- Dosya Paylaşma
- Paylaşılan dosyayı indirme
- Paylaşılan dosyada düzenleme yapma

- 1) Şifre değişim talebi: Kullanıcı şifre değişiminde bulunmak için sistem yöneticisine talepte bulunur. Eğer sistem yöneticisi talebi onaylarsa “set new password” alanından şifresini değiştirebilir. Eğer sistem yöneticisi talebi reddederse , ilgili kullanıcı sisteme girdiğinde talebin reddedildiğine dair bildirim alır.

İlgili Fonksiyonlar:

- a) **create_password_reset_request():** Kullanıcı Request Password Reset butonuna bastığında bu fonksiyon tetiklenir ve kullanıcı bu sayede sistem yöneticisine şifre değişikliği talebinde bulunabilir.
 - b) **check_reset_request_status():** Bu fonksiyon, talebin durumunu kontrol eder. Eğer şifre değişim talebi onaylanmış ise bireysel kullanıcının arayüzüne “Set New Password” isimli bir buton çıkar ve bu buton ile yeni şifresini girerek kayıt eder.
 - c) **set_new_password():** Bu fonksiyon ile kullanıcının yeni girilen şifresi kayıt edilir.
- 2) Kullanıcı Adı Değişimi: Kullanıcı, kullanıcı adı bilgisini istediği zaman değiştirebilir.

İlgili Fonksiyonlar:

- a) **change_username():** bu fonksiyon ile kullanıcı yeni username bilgisini girer.
- b) **save_new_username():** Bu fonksiyon ile girilen yeni username bilgisi kayıt edilir ve artık kullanıcının usernami, yeni girilen usernamedir.

- 3) Sisteme Dosya Yükleme: Kullanıcı sisteme istediği dosyayı yükleyebilmelidir. Bu işlem, kullanıcının toplam dosya yükleme limiti dolana kadar devam edebilir. Depolama limiti dolduğunda veya toplam depolama limitini aşacak bir dosya yüklendiğinde kullanıcıya uyarı bildirimi gönderilmiştir.

İlgili Fonksiyonlar:

- a) **upload_file_ui():** Bu fonksiyon ile **upload_file()** dosyası çalıştırılır ve kullanıcıya bir arayüz sunulur.
 - b) **upload_file():** Bu fonksiyon ile kullanıcı sisteme dosya yükleyebilir. Eğer yeni yükleyeceği dosyanın ismi, kullanıcının dosyalarında zaten mevcutsa o isimde yeni bir dosya yükleyemez. Kullanıcı başarılı bir şekilde dosya yükleme işlemini gerçekleştirdiğinde yeni dosyanın büyüklüğü hesaplanarak kullanıcının mevcut depolama boyutu güncellenir.
 - c) **log_upload_file_activity():** Bu fonksiyon ile kullanıcının başarılı bir şekilde dosya yükleme durumunda userlog.txt dosyasına zaman ve kullanıcı bilgileri ile birlikte dosyanın yüklenmesi ile ilgili bilgi yazılır. Bu fonksiyon **upload_file()** fonksiyonunda kullanılır.
- 4) Üye Ekleme: Kullanıcı dosya alışverişinde bulunabileceği üye ekleyebilir.

İlgili Fonksiyonlar:

- a) **add_user_relations():** Bu fonksiyon ile kullanıcının üye eklemesi sağlanır. Bu sayede kullanıcı ve eklene kişi arasında bir ilişki oluşturulur.

- b) `get_username()` Kullanıcı adı bilgisi alınır.
- c) `notify_user()` Bu fonksiyon ile ilişki kurulan kullanıcıya bildirim gönderilir.
- d) `list_related_users()` Bu fonksiyon ile kullanıcının ilişkili olduğu kişiler kullanıcıya gösterilir.
- e) `log_user_activiy:` Bu fonksiyon ile kullanıcı bir kişiyi üye olarak eklediğinde `userlog` dosyasına eylem yazılır.

yazılır. Bu fonksiyon, `download_file` fonksiyonunda çağrılır.

- 7) Dosya Düzenleme: Kullanıcı kendisine gönderilen dosyanın adını değiştirebilir.

İlgili Fonksiyonlar:

- a) `rename_file_ui()`: Kullanıcı tarafından gönderilen dosya bu fonksiyon ile isim değişikliğinde bulunulabilir.

- 5) Dosya Paylaşma: Kullanıcı üye olarak eklediği kişilerde dosya paylaşımında bulunabilir. Dosya göndermek istediği kişiyi `my members` bölümünden seçer ve `share file` butonu ile kendi dosyalarından dosya seçer. Daha sonra `send file` butonu ile ilgili dosya ilgili kullanıcıya gönderilir.

- a) `show_my_members_ui()`: Bu fonksiyon ile kullanıcının ilişkili olduğu kişiler listelenir ve her kullanıcının yanında `share file` butonu bulunur.
- b) `show_share_files_ui()` bu fonksiyon ile kullanıcının paylaşabileceği dosyalar listelenir. Her bir dosyanın yanında `send file` butonu bulunur.
- c) `send_file()`: Bu fonksiyon ile kullanıcı istediği dosyayı gönderebilir.

- 6) Paylaşılan Dosyayı İndirme: Kullanıcı kendisine gönderilen dosyaları indirebilir.İndirdiği dosya artık kendisinin dosyası olur ve istediği kişiyle paylaşır.

- a) `show_received_file_ui()` Bu fonksiyon ile kullanıcıya gönderilen dosyalar, gönderen bilgisi ile listelenir. Her dosyanın yanında `download` butonu bulunur ve bu buton aracılığıyla paylaşılan dosyalar indirilir.
- b) `download_file()`: Bu fonksiyon aracılığıyla kullanıcı kendisine gelen dosyalar indirilir.
- c) `log_download_file_activity()`: Kullanıcının başarılı bir şekilde dosya indirmesi durumunda `userlog` dosyasına bu eylem saat ve `username` bilgisi ile birlikte

III.Kullanışı Davranışı Analiz Modülü

Kullanıcının davranışları analiz edilir ve `userlog` dosyasına bu davranışlar yazılır. Bu davranışların tarihi, seviyesi (WARNING veya INFO) ve yapılan eylem `userlog` dosyasına yazılır. Kullanıcının, giriş işlemleri, şifre talebi işlemleri, dosya yükleme işlemleri, dosya indirme işlemleri, dosya paylaşım işlemleri ve kullanıcının arkadaş (üye) ekleme işlemleri `userlog` dosyasına yazılır.

1) Giriş davranışları

Kullanıcı giriş işlemi yaptığında `userlog` dosyasına şu şekilde örnek bir yazı yazılır: 2024-12-29 16:50:16 -INFO- fatih logged in to system. Burada fatih kullanıcısının giriş yaptığı saat, level durumu ve aksiyon yazılmıştır.

2) Hatalı Giriş İşlemi

Kullanıcı Hatalı bir giriş işleminde bulunması durumunda `userlog`a şuna benzer bir mesaj yazılır: “2024-12-29 18:09:57 - WARNING- fatih kullanıcısı hatalı giriş işleminde bulundu.”

3) Şifre değişim talebi işlemleri

Kullanıcının şifre değişim talebinde şuna benzer bir metin bilgisi `userlog` dosyasına yazılır: “2024-12-29 15:48:30 -INFO- fatih requested a password reset.” Burada fatih kullanıcısının talebi yaptığı tarih-saat bilgisi ile birlikte yaptığı eylem yazılmıştır.

4) Dosya yükleme işlemi

Kullanıcının dosya yükleme durumunda şuna benzer bir mesaj userlog dosyasına yazılır: “2024-12-29 16:05:29 -INFO- fatih Position_Salaries.csv adlı dosyayı yükledi.”. Burada fatih kullanıcısının dosyayı yüklediği tarih-saat bilgisi ile yaptığı eylem yazılır.

5) Arkadaş Ekleme İşlemi

Kullanıcı tarafından arkadaş eklendiğinde userlog dosyasına şuna benzer bir mesaj yazılır: “2024-12-29 16:10:56 - INFO – fatih timuru arkadaş olarak ekledi.”. Burada arkadaş ekleyen ve eklenen kişinin kullanıcı isimleri , tarih-saat bilgisi ile birlikte mesaj olarak eklenir.

6) Dosya Paylaşım İşlemi

Kullanıcı tarafından dosya paylaşıldığı zaman userlog dosyasına şuna benzer bir şekilde mesaj yazılır: “2024-12-29 16:09:13 - INFO - fatih sent file 'asdf.txt' to timur”. Burada dosya gönderen ve gönderilen kişilerin username bilgileri, tarih-saat bilgisi ile birlikte mesaj olarak yazılır.

7) Dosya İndirme İşlemi

Kullanıcı kendisine gönderilen bir dosyayı indirdiğinde userlog dosyasına şuna benzer bir mesaj yazılır: “2024-12-29 16:11:22 - INFO- fatih timur tarafından gönderilen abc.txt dosyasını indirdi.”. Burada dosyayı indiren ve gönderen kişilerin username bilgileri, tarih-saat bilgileri ile birlikte yazılmıştır.

Anormal durumlar sistem yöneticisine bildirim olarak gönderilir. Sistem yöneticisi, her bir anormal durumun içeriğini tarih-saat bilgisi ile birlikte görebilir.

A) Anormal Durumlar

- Kullanıcının belirli bir süre zarfında 3 den fazla hatalı giriş denemesinde bulunması.
- Kullanıcının belirli bir zaman zarfında olağandışı dosya yüklemesi.
- Kullanıcının belirli bir zaman zarfında olağandışı dosya indirmesi.
- Kullanıcının belirli bir zaman zarfında olağandışı şifre değişim talebinde bulunması.

B) İlgili Fonksiyonlar

1) read_state()

Bu fonksiyon, state.txt dosyasından son işlenmiş log satırının numarasını (last_processed_line) okur. Eğer dosya mevcut değilse veya dosya içeriği geçerli bir sayı değilse, varsayılan olarak 0 döndürür. Bu, işlemeye başlanacak log dosyasının hangi satırdan itibaren okunacağını belirlemek için kullanılır.

2) write_state(last_line)

Son işlenen log satır numarasını state.txt dosyasına yazar. Bu, bir sonraki çalışmada kaldığı yerden devam edebilmek için durum bilgisini kaydeder.

3) read_log_file()

Log dosyasındaki tüm verileri satır satır okur ve bir liste olarak döndürür. Bu liste, anomali tespiti işlemleri için kullanılır.

4) write_anomaly_log(anomalies)

Yeni tespit edilen anomalileri anomaly_log.txt dosyasına zaman damgasıyla birlikte yazar. Önce mevcut anomaliler okunur ve üzerine yeni tespit edilen anomaliler eklenir.

5) read_anomaly_log()

anomaly_log.txt dosyasındaki mevcut anomalileri okur ve bir liste olarak döndürür.

IV. Log İzleme ve Anormallik Tespiti Modülü

Sistemdeki anormal durumlar, userlog dosyası thread aracılığıyla satır satır okunarak tespit edilir. Anormal durumlar, log dosyası okunurken anahtar kelimeler kullanılarak bulunur. Tespit edilen bu anormal durumlar, daha sonra okunmak için “ anomaly_log” dosyasına yazılır.

Eğer dosya mevcut değilse, boş bir liste döndürülür.

göstergesi olabileceğinden, eşik aşılsa anomali olarak kaydedilir.

6) **detect_anomalies()**

Bu fonksiyon, log dosyasındaki yeni satırları okuyarak anomali tespiti yapar. Aşağıdaki alt kontrolleri çağırır:

- **check_failed_logins:** Başarısız giriş denemeleri
- **check_multiple_reset_requests:** Şifre sıfırlama talepleri
- **check_multiple_file_uploads:** Dosya yükleme işlemleri
- **check_multiple_file_downloads:** Dosya indirme işlemleri
- Tespit edilen anomaliler log dosyasına yazılır ve işlenen son satır numarası kaydedilir.

7) **check_failed_logins(log_lines, time_window=180, max_attempts=3)**

Bu fonksiyon, kullanıcıların kısa bir süre içinde (örn. 3 dakika) ardışık başarısız giriş denemelerini kontrol eder. Belirli bir eşiği aşan giriş denemeleri anomali olarak işaretlenir ve raporlanır.

8) **check_multiple_reset_requests(log_lines, time_window=180)**

Bu fonksiyon, bir kullanıcının kısa bir süre içinde (örn. 3 dakika) birden fazla şifre sıfırlama talebinde bulunup bulunmadığını kontrol eder. Belirtilen eşiği aşan talepler anomali olarak kaydedilir.

9) **check_multiple_file_uploads(log_lines, time_window=60)**

Bir kullanıcının bir dakika içinde üçten fazla dosya yükleyip yüklediğini kontrol eder. Belirtilen eşiği aşan yükleme işlemleri anomali olarak raporlanır.

10) **check_multiple_file_downloads(log_lines, time_window=60, max_downloads=3)**

Bir kullanıcının bir dakika içinde üçten fazla dosya indirme işlemi yapıp yapmadığını kontrol eder. Bu işlem bir güvenlik ihlali

11) **get_new_log_lines(log_lines)**

Log dosyasındaki henüz işlenmemiş yeni satırları döndürür. `last_processed_line` global değişkenini günceller.

12) **run_anomaly_detection()**

Anomali tespit sürecini başlatır ve periyodik olarak log dosyasını tarar. Sürekli çalışan bir döngü içinde her 5 saniyede bir `detect_anomalies()` fonksiyonunu çağırır.

13) **start_anomaly_detection()**

Bu fonksiyon anormal durumların tespiti için `run_anomaly_detection()` fonksiyonunu thread aracılığı ile başlatır

14) **add_anomaly_notifications()**

Bu fonksiyon sistem çalıştığında otomatik olarak çalışır ve tespit edilen anormal durum mesajları admine gönderilir.

V. KAYNAKÇA

[2] <https://www.youtube.com/watch?v=gdIdPbphA7g>

[3] https://www.youtube.com/playlist?list=PLSmHiN0iazy_qX_6Tmecj4tTOefqh2-m2

[4] <https://www.geeksforgeeks.org/multithreading-python-set-1/>