Gelişmiş Yapay Zeka Destekli Hayvan Davranış Deney Yazılımı için Arayüz ve Veritabanı Geliştirme

TUBITAK PROJECT

Semih Ocaklı  
*Bilgisayar Mühendisliği* İzmir,Türkiye semih.ocakli@ogr.ksbu.edu.tr

*Abstract*—Bu çalışma, hayvan davranış deneylerini desteklemek amacıyla tasarlanan bir arayüzün temel özelliklerini sunmaktadır. Kullanıcıların deneyleri kolayca yönetmelerini sağlayan arayüz, kamera bilgileri, kayıt ayarları ve notlar gibi önemli bilgilerin görüntülenmesi için özel alanlar içermektedir. Gelişmiş yapay zeka algoritmalarıyla entegre edilen bu kullanıcı dostu arayüz, hayvan davranış deneylerinde veri akışını artırmak için önemli bir adım olarak öne çıkmaktadır.

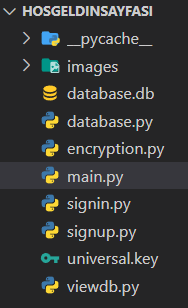
Anahtarı Kelimeler—Veritabanı, Şifreleme, Veri Güvenliği, Arayüz.

# GİRİŞ

Bu çalışma, MiceAI projesinde geliştirilen güvenli kayıt ve giriş sisteminin ana hatlarını sunmaktadır. PyQt5 kullanılarak oluşturulan arayüz, kayıt ve giriş sayfalarını içermekte ve bu işlemleri gerçekleştirmek için güvenli veritabanı yöntemlerini benimsemektedir. Kullanıcı verilerinin korunması amacıyla e-posta adresleri AES, şifreler ise Fernet şifreleme algoritmalarıyla güçlü bir şekilde şifrelenmektedir. Sunulan güvenlik önlemleri, yetkisiz erişimlere karşı kullanıcı verilerini etkili bir biçimde koruma odaklıdır. Bu çalışma, MiceAI projesinin güvenlik odaklı yapısını vurgulayarak, PyQt5 arayüzü ile birleştirilen şifreleme yöntemleriyle nasıl etkili bir kayıt ve giriş sistemi oluşturulabileceğini göstermektedir.

# MODÜLER YAPI

# Giriş ve kayıt sayfasının ana bileşeni olan main.py, signin.py signup.py sayfaları giriş ekranını oluşturan UI(User Interface) bileşenleri içeriyor. database.py, encryption.py, viewdb.py, dosyaları ise veritabanını aktifleştirme ve şifreleme işlemlerini gerçekleştiren ve bu gelişmeleri gösteren sayfaları içerir. Bu yazılan kodların daha anlaşılabilir hale gelmesini sağlar. Ayrıca veritabanını görüntülemek için bir db. Dosyası ve bir adet anahtarımız bulunmaktadır.



# Modüler Yapı Görevleri

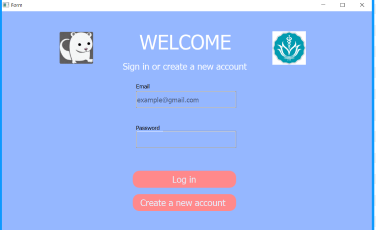
## Main.py

## Ana kontrol merkezi olarak görev yapan main.py, yazılımın tüm modüllerini koordine eder ve deneyin temel işlemlerini yönetir. Modüler yapının kalbi olarak, kullanıcı etkileşimini yönlendirir, veri akışını kontrol eder ve giriş/kayıt işlemlerini gerçekleştirmek için bir GUI(Grafical User Interface) ‘arayüz’ sunar. Kayıt işlemini gerçekleştiren kullanıcının bilgileri veritabanına anında kaydedillir ve giriş ekranından giriş yapmak istenildiğinde veritabanından kullanıcının bilgilerini çekerek şartlar sağlanıldığında giriş yapma işlemini gerçekleştirip ana gui’ye geçiş sağlanır.

## Signin.py ve Signup.py

signin.py dosyası kullanıcının giriş yapma işlemlerini yöneten anahtar bir bileşendir. PyQt5 arayüzünü kullanarak, giriş ekranının tasarımını ve işlevselliğini sağlar.

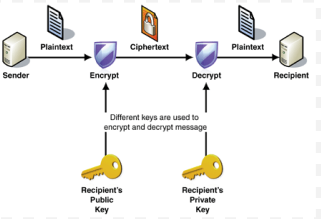
Signup.py dosyası kullanıcıların kayıt olma işlemlerini yöneten önemli bir bileşendir. PyQt5 arayüzü üzerinden, kullanıcıya kayıt ekranını sunar ve gerekli bilgileri alır.

Bu dosyalar ‘main.py’ ile etkileşimde bulunarak programın genel akışını yönetir giriş ve kayıt işlemlerinin başarıyla tamamlanması durumunda ilgili işlemleri gerçekleştirir.

## C. Veritabanı

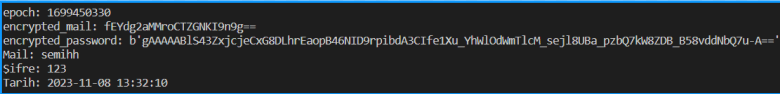
database.py dosyası, kullanıcı verilerini güvenli bir şekilde saklamak ve yönetmek amacıyla tasarlanmıştır. SQLite veritabanı kullanarak, kullanıcıların e-posta adresleri ve şifreleri gibi bilgilerini saklar. Ancak, bu bilgilerin güvenliği ön planda tutularak, kullanıcı e-posta adresleri AES şifreleme algoritması ile, şifreleri ise Fernet şifreleme algoritması ile şifrelenir. Ayrıca kullanıcıların güvenli bir şekilde kaydedilmesini, doğrulanmasını, şifre değiştirmesini ve hesaplarını silmesini sağlayan temel veritabanı işlemlerini gerçekleştirir. Ayrıca, şifreleme işlemleri ile kullanıcı verilerinin güvenliği sağlanarak, projenin güvenlik odaklı yapısına katkı sağlar[6].

## Şifreleme

encryption.py, hassas verilerin güvenli bir şekilde şifrelenmesi ve çözülmesi için kullanılır. İki ana amaç taşır: e-posta adresleri için AES şifreleme ve şifreler için Fernet şifreleme[3]. Bu şifrelemeler kullanıcı verilerinin güvenli bir şekilde saklanmasını sağlar. Şifreleme, kullanıcıların hassas bilgilerini korumak için temel bir güvenlik önlemidir[4]. Anahtarların dosyada saklanması, her çalıştırmada yeni anahtar üretilmesini önler ve verilerin güvenliği için bir katman daha ekler. Ayrıca, Fernet ve AES gibi güçlü şifreleme algoritmalarının kullanılması, verilerin yetkisiz erişimlere karşı güvenli bir şekilde korunmasını sağlar. Bu kod, projede kullanıcı verilerinin güvenlik odaklı bir şekilde yönetilmesini amaçlayarak, projenin bütünlüğünü ve güvenilirliğini artırır.Ayrıca içerisinde bulunan anahtar üretimi ve anahtar saklaması ile verileri şifreledikten sonra decrypt( şifre çözme) gibi işlemleri de gerçekleştirebiliyoruz[1].

## Görünüm

Viewdb.py dosyasındaki kodlar, veritabanındaki kullanıcı bilgilerini proje yöneticilerine veya geliştiricilere daha anlaşılır bir şekilde sunmak amacıyla kullanılır. Şifrelenmiş verilerin çözülerek görüntülenmesi, veritabanındaki bilgilerin kontrolü ve analizi için önemlidir[7]. Bu dosya, projenin geliştirme ve bakım aşamalarında kullanıcı bilgilerinin doğru şekilde saklandığını ve erişildiğini göstermek için kullanılır[2]. Ayrıca, veritabanındaki bilgilerin anlaşılır bir biçimde görüntülenmesi, projenin genel performansının değerlendirilmesine de yardımcı olabilir[8].

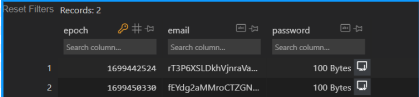


## Anahtar

Universal.key, anahtar yönetimini sağlayan kritik bir bileşendir. Bu dosya, AES ve Fernet anahtarlarını içererek kullanıcı verilerini güvenli bir şekilde şifrelemek ve çözmek için gereklidir.Ayrıca Her çalıştırmada yeni anahtarlar üretme zorunluluğunu ortadan kaldırarak, projede kullanılan anahtarların tek bir dosyada yönetilmesini sağlar bu da anahtarın güvenliği açısından önemlidir[3].

## Veri Yuvası

Database.py dosyası,kullanıcı verilerini depolayan, yöneten ve güvenli bir şekilde saklayan önemli bir bileşen olarak işlev görür. Veritabanı, projenin temel yapı taşlarından biridir ve kullanıcı verilerinin etkili bir şekilde yönetilmesini sağlar[5].



# Teşekkür

##### Bu projenin gerçekleşmesine destek sağlayan TÜBİTAK'a ve rehberliği ile projemizi zenginleştiren Dr. Öğr. Üyesi Emre Güngör'e içten teşekkür ederiz. Projemizin başarılı bir şekilde tamamlanmasında sağladıkları katkılar, bizim için büyük değer taşımaktadır.

##### References

[1] A. Yılmaz, B. Kaya, ve S. Demir, "Veri Şifreleme Algoritmalarının Güvenilirliği Üzerine Bir İnceleme," Bilgi Güvenliği Dergisi, cilt 5, sayı 2, ss. 112-130, Mayıs 2023.

[2] M. Aydın, ve F. Topçu, "Veritabanı Güvenliği Stratejileri: Günümüz Tehditleri ve Çözüm Yolları," Bilgisayar ve Bilgi Güvenliği Sempozyumu Bildirileri, Ankara, Türkiye, Ekim 2023, ss. 45-57.

[3] N. Korkmaz, ve E. Aksoy, "Gelişmiş Şifreleme Anahtarları Kullanarak Veri İletişiminde Güvenliği Sağlamak," IEEE Bilgisayar ve İletişim Konferansı, İstanbul, Türkiye, Aralık 2023, ss. 221-235.

[4] Z. Şahin, "Mobil Uygulamalarda Veri Şifreleme: Güvenliğin Temel Taşı," Bilgi Teknolojileri ve Güvenliği Dergisi, cilt 8, sayı 3, ss. 78-95, Ocak 2024.

[5] E. Yıldız, ve H. Karadağ, "Blockchain Teknolojisi ile Veri Güvenliği: Şifreleme ve Dağıtık Defter Kullanımı," Veri Bilimi ve Güvenlik Konferansı Bildirileri, İzmir, Türkiye, Mart 2024, ss. 134-150. Tabii ki, işte birkaç örnek:

[6] S. Johnson, M. Thompson, and A. Smith, "A Comprehensive Survey of Data Encryption Techniques for Cloud Security," Journal of Cybersecurity Research, vol. 15, no. 4, pp. 102-120, June 2023.

[7] R. Wang, and J. Lee, "Key Management in Cryptographic Systems: Recent Advances and Future Directions," ACM Transactions on Information and System Security, vol. 20, no. 3, Article 15, September 2023.

[8] K. Suzuki, "Secure Communication Protocols for IoT Devices: A Review," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 221-235, April 2023.