Python Programlama Şifre Takip Programı (Password Guard)

Özet

Uygulamamız konsol uygulamasıdır. Bunun yanında görsel arayüz oluşturmak için harcanacak vakti, kodların daha profesyonel olması için katmanlı mimari kullanılmaya çalışılmıştır. Bu sayede kodların okuması ve karmaşıklığı azalması amaçlanmaktadır.

Uygulamamızda Sqlite veri tabanı kullanılmıştır. Şifreler encryption yapılarak veri tabanına eklenmiş ve discryption işlemi yapılarak veri tabanından çekilmiştir.

Uygulamaya ilk girildiğinde bir kullanıcı hesabı oluşturulduktan sonra daha sonraki girişlerde kullanıcı, oluşturduğu şifre ile hesaba giriş yapabilmektedir.

Uygulamaya giriş yapan kullanıcı; eski kayıtlarını görebilir, web sitesi şifresini güncelleyebilir ya da yeni bir şifre oluşturabilir. Yeni şifre oluşturma işlemi ise kısaca şöyledir: “Web sitesi adını”, “web sitesi adresini” ve “web sitesi şifresini” kısımlarını doldurarak uygulamamızda yeni bir şifre oluşturabilmektedir.

Kullanıcı daha önce kaydetmiş olduğu bir siteyi, sitenin adını yazarak ulaşabilmektedir.

Şifresini güncellemek isteyen bir kullanıcı ise site adını ve yeni şifreyi girerek şifresini güncelleyebilir.

Şifreler veri tabanına kaydedilirken şifrelenerek kaydedilir. Kullanıcı tekrar şifreyi görmek istediğinde şifre çözülerek kullanıcıya gösterilir.

Giriş

1) Veri tabanı ve yapısı:

Uygulama için kullanılan veri tabanı Sqlite veri tabanıdır. Veri tabanımız 3 adet tablodan oluşmaktadır.[Şekil 1]



**Şekil 1: Tablolar**

1.1) USER Tablosu:

Bu tabloda uygulamaya kayıt olmuş kullanıcıların kullanıcı adları ve şifreleri tutulmaktadır.[Şekil 2]



**Şekil 2: USER Tablosu**

1.2) ACCOUNT Tablosu:

Veri tabanımızdaki “ACCOUNT” tablosu ile uygulamaya girilen “site adları”, “site adresleri” ve “site şifresi” bilgileri bu tabloda tutulmaktadır. “USER\_ID” sütunu sayesinde ise bu bilgiler hangi kullanıcının olduğu bilgisi tutulmaktadır.[]



**Şekil 3: ACCOUNT Tablosu**

Şekil 3’de görüldüğü üzere şifreler, şifrelenmiş şekilde veri tabanına kaydedilmiştir. Bu veriyi çekerken şifre çözülerek kullanıcıya oluşturduğu şifre olarak gönderilmektedir. Bu yapının nasıl oluşturulduğu Uygulama kod yapısındaki OdevPytHashHelper bölümünde anlatılmıştır.

1.3) sqlite\_sequence Tablosu:

Bu tablo Sqlite veri tabanının kendisinin oluşturduğu ve tablolarda kaç tane satırın bulunduğunu tutan bir tablodur.



**Şekil 4: sqlite\_sequence Tablosu**

2) Uygulama Kod yapısı:

2.1) Main classı:

Uygulamanın çalıştığı sınıf burasıdır. Bütün işlemler fonksiyon kullanılarak yapılmıştır. Her işlemi tek bir sayfaya yazmak yerine her işlem farklı classlara ayrılmış ve bu classlar main classın import edilerek kullanılmıştır. Kullanıcı yönlendirmek için kullanılan sorular ve sorular karşısında kullanıcının verdiği cevaplara göre uygulamanın hareket etmesi bu sınıf içinde sağlanmıştır.

2.2) Account ve User classı:

Bu sınıflar Sqlite veri tabanımızdaki tabloların birer yansımasıdır. Yani Mvc modelimizdeki “Model” kısmımızdır. Veri tabanımızdaki tablolar için ihtiyacımız olan bilgilerin değişkenleri burada tutulur.

2.3) OdevPytData classı:

Bu class veri tabanı bağlantısını yapmamızı sağlar ve bu bağlantı ile sorgularımızı çalıştırmaya yarar.

2.4) OdevPytController classı:

Bu sınıf sayesinde OdevPytData classında yaptığımız işlemleri yönetme imkanı buluruz. Bunun yanı sıra şifreleri hashlememize yarayan odevPytHashHelper sınıfı da bu sınıf içine import edilmiştir.

2.5) OdevPytHashHelper classı:

Bu sınıfta kullanıcıdan aldığımız verileri şifrelemiştir. Aynı zamanda veri tabanından veri çekerken şifrelediğimiz verileri tekrar çözümlemeyi sağlanmıştır.

2.6) Kullanıcılan kütüphaneler:

|  |  |  |
| --- | --- | --- |
| **Class** | **IMPORT** | **AÇIKLAMA** |
| User.py | import işlemi olmamıştır |  |
| Account.py | import işlemi olmamıştır |  |
| odevPytHashHelper.py | import base64 |  |
| odevPytData.py | import sqlite3 as sql  import Account | * Sqlite’ı bağlamak için sqlite3 kütüphanesi kullanılmıştır. * Account classı ile işlem yapılacağı için bu class’da import edilmiştir |
| OdevPytController.py | import odevPytData as data  import odevPytHashHelper as hash  import User  import Account | * Sadece kullanılacak classlar import edildi. Harici bir kütüphane kullanılmadı |
| Main.py | import odevPytController as control  import User  import Account | * Sadece kullanılacak classlar import edildi. Harici bir kütüphane kullanılmadı |

3) Uygulama Görünümü ve kullanımı:

Uygulamayı açtıktan sonra kayıt olma ya da giriş yapma işlemlerinin olduğu bir kısım gelmektedir. “1” e basarak giriş yapabilir. “2” ye basarak kayıt olunur. [Şekil 5]



**Şekil 5**

“2”ye bastık ve yeni bir kullanıcı oluşturmak istenildiğinde karşımıza [Şekil 6]daki gibi ekran gelmektedir. Buradan kullanıcı adı ve şifre oluşturularak kayıt yapılmaktadır.



**Şekil 6**

Kullanıcı adı ve şifremizi “sunum” olarak verdik. Kayıt olduktan sonra Şekil[7]deki gibi ekran gelmektedir.



**Şekil 7**

[Şekil 7]’de üç tane seçenek bulunmaktadır. Kullanıcı “1” numarayı seçerse kaydetmiş olduğu şifrelerin adını yazdığını görebilir. [Şekil 9]

Kullanıcı “2” numarayı seçerse Yeni bir hesap ve şifre oluşturabilir. [Şekil 8]

Kullanıcı “3” numarayı seçerse girmiş olduğu sitenin şifresini değiştirebilir. [Şekil 10]



**Şekil 8: Hesap ekleme**



**Şekil 9: Hesap Arama**



**Şekil 10: Şifre Güncelleme**

Şifremizi güncelledikten sonra değişen şifreye tekrar göz atalım.



**Şekil 11: Şifre güncelleme sonuç**