Universitatea
Transilvania
din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

PROIECT ISW

Conducător științific:

Ciobanu Cătălin

Studenți:

Grupa 4LF692:

Teodorescu Adrian

Pascu Flavian

Grupa 4LF691:

Hozu Marius

Manole Alexandru

BRAȘOV, 2023

Departamentul Electronică și Calculatoare

Programul de studii: Tehnologii și Sisteme de Telecomunicații

Manager-Teodorescu Adrian-Iulian,

Documetație-Pascu Flavian

Developer-Manole Alexandru

Tester-Hozu Marius

PASSWORD MANAGER

Conducător științific:

Ciobanu Cătălin

Brașov, 2023

Sumar

Obiectivul proiectului

Obiectivul proiectului nostru este să dezvoltăm un Password Manager, capabil sa retina intr-o baza de date locala parlolele introduse de utilizator și sa genereze prin limbajul de criptare SHA-256 o cheie pentru resetarea bazei de date.

Funcționalități cheie

- Generare parole puternice
- Stocarea securizată a parolelor
- Interfața intuitiva
- Protecție cu parola de master
- Resetarea parolei

Beneficii

- Securitatea parolelor
- Ușurința utilizării
- Economisirea timpului
- Protectie împotriva atacutilor de phishing
- Protecție împotriva keyloggers

Table of Contents

Sumar 2

1.Introducerea 4

1.1 Ce este un password manager? 4

1.2 Modalitati de creeare a parolei 6

1.3 Criptografia 7

14 Criptarea cu cheie publica si cheie private 8

1.5 Hashing 10

2.Descrierea Aplicatie 12

3.Tutorial de folosire a aplicatiei 13

4.Tehnologii folosite 16

5.Testare 21

6.Management 23

7. GITHUB REPOSITORY 25

8.Lectii invatate 26

9.Concluzii 27

10.Bibligrafie 30

1.Introducerea

1.1 Ce este un password manager?

Un manager de parole este o aplicație software care este utilizată pentru a stoca și gestiona parolele pe care le are un utilizator pentru diferite conturi online și funcții de securitate.

Managerii stochează parolele într-o forma criptat oferind acces securizat la toate informațiile despre parole cu ajutorul unei parole principale.

Parolele reprezintă autentificarea în sistemul informatic sau web. Din punctul de vedere a utilizatorilor, memorarea unei parole unice este uşor de gestionat , mai multor parole.

Pentru rau voitori, utilizarea unei singure parole insemaan ca sistemele vor fi in mod automat compromise, daca parola dintr-un system slab protejat va fi sparta cu success.

Parolele asigurand prima linie de aparare inpotirva accesului neautorizat.

Pentru realizarea unui management mai efficient si securitatea parolellor se recomanda:

O parola nu trebuie sa semene cu codul numeric personal, data nastii, numarul de teleofn, numele,prenumele, login-ul etc, cu numele strazii, modelul masinii, sloganul unor organizatii, sa fie cuvinte din dictionar sau termini tehnici.

Parolele nu trebuie impartasite nimanui, nu conteaza de situatie, chiar si cu scopul mentenantei periodice a calculatorului in cazuri de genul creeaza un cont nou cu acces corespunzator.

Parolele commune trebuie inlocuite imediat ce o persoana iese din grup sau nu mai are accesul de utilizator.

Parolele trebuie inlocuite imediat ce se constata unele banuieli referitoare la cunoasterea lor

Parolele trebuie tinute daca se poate minte. Daca este prea greu de tinut minte mai multe parole, un password manager paote fi solutia buna

Oprirea operatiilor de incercare repetata de logare. Niciodata sa nu se introduca parola dupa ce a fost urmata de un link dintr-un email primit si care nu este de incredere

Daca un terminal functionaza pentru o perioada lunga de timp, procesul de inregistrare trebuie sa aiba loc la intervale regulate de timp pentru asigurarea sistemului ca nu este folosit de altcineva .

La creearea unei noi sesiuni, utilizatorul trebuie sa ii fie adus la cunostinta ultima data cand sistemul a fost accesat cu aceea parola, evitanduse reutilizarea parolei anterioare.

Daca contul de utilizator a fost compromise anterior, fie inconstient sau constient, utilizarea parolei inmmod repetat ar putea compromite

De asemenea, în cazul care o parolă a fost divulgata pentru un oarecare motiv, reutilizarea ei ar putea permite cuiva acces neautorizat la conturi. Evitarea utilizării aceleași parole pentru mai multe conturi.

- Să nu se utilizată functonalitatea de logare automată. Aceasta diminuează valoarea parolei, dacă un inamic va avea acces la un sistem configurat, el va putea să preia controlul asupra întregului sistem și a avea acces la informații potențial sensibile.

Notificarea despre schimbarea parolei.

Actualizarea adresei de e-mail de recuperare, pentru a putea primi emailuri în cazul în care este nevoie.

De asemenea, se adăuga un număr de telefon pentru primirea codurile de resetare a parolei prin mesaj.

În plus, multe site oferă posibilitatea de a răspunde la o întrebare de securitate în cazul în care parola se uitată. Dacă întrebarea este creată deutilizator,, răspunsul la ea ar trebui să fie cunoscut doar de utilizator. Răspunsul nu ar trebui să poată fi ghicit din informațiile pe care sunt postate on-line pe profilurile din rețelele sociale.

Dacă întrebarea este aleasă din o listă de opțiuni, cum ar fi locul nășterii , răspunsul trebuie personalizat astfel încat, chiar dacă cineva nimereste răspunsul, nu va ști cum să introducă în mod correct raspunsul. Sistemul de calcul nu trebuie lăsate nesupravegheat fără a activa un sistem de blocare a accesului la acestea; deblocarea trebuie să se facă folosind parola

O parolă puternică poate reduce riscul nimeririi sau aflării sale prin atacul brute.Forta unei parole este determinată de lungimea, complexitatea şi impredictibilitatea aparitiei caracterelor în componenta sa.

Pentru crearea unei parole puternice este recomandat să conțină:

- cel putin 16 caractere lungime şi nu 6

- litere mari și mici (A-Z, a-z);

- cel puțin o cifra (0-9);

- cel puțin un semn special (~! @ # $% ^ & * () _- + =);

1.2 Modalitati de creeare a parolei

Modalitatile de creeare a parolei sunt fie prin generarea aleatoare fie prin construirea ei de catre utilizatori.

Generarea de parole aleatoare fiind o metoda recomandata pentru utilizare, aplicatiile de acest tip implementand recomadari generale pentru alcatuirea unei parole puternice, dar greu de tinut minte

Utilizatorii alegand o parola folosindu-se de propia gandire, dar in cele mai multe cazuri acestea utilizeaza cuvinte integi din dictionar sau parti din acesta, serii de caractere,etc.

Desi fiind usor de memorat, parolele create sunt usor de ghicit, alegerea unei astfel de parole find o greseala mare.

Pentru creearea de parole puternice si usor de memorat s-au inventat mai multe metode, metoda mnemonicii , selecatandu se o fraza si extregandu-se cate un caracter din fiecare cuvant.

Frazele care sunt transformate in parole mnemonice, fara folosirea substitutiei de caractere sau alte modificari, pot fi usor de ghicit de catre atacatori.

Utilizatorii care compun astfel de parole trebuie sa creeze propiile fraze, sau sa se faca un schimb neasteptat , cum fi utilizarea literelor mari, a simbolurilor si redarea complete a unor cuvinte.

Fraze modificate

- Selectarea unei fraze si modificarea formei intr-o derivate. Aceasta permite creearea parolelor de dimensiuni mari, complexe si simple de memorat,

Modificarea cuvintelor si combinarea

- Combinarea a doua sau trei cuvinte fara legatura intre ele si schimbarea unei litere cu caractere speciale sau litere.

1.3 Criptografia

Cuvantul criptografie are origine greaca > kryptos=ascuns; graphein=scriere.Este o scriere codificata in care doar emitatatorul si receptorul mesajului il poate interpreta. Acesta este unul dintre principalele mecanisme de Securitate digitala utlilizate pentru a proteja datele si informatiile confidentiale de amenintarile legate de utilizarea Internetului.

Criptarea face posibila transformarea unui mesaj citibil intr-unul invizibil

Comunicarea sigură se referă la scenariul în care mesajul sau datele partajate între două părți nu pot fi accesate de un adversar.

În Criptografie, un Adversar este o entitate rău intenționată, care își propune să recupereze informații sau date prețioase, subminând astfel principiile securității informațiilor.

Confidențialitatea datelor, integritatea datelor, autentificarea și non-repudierea sunt principiile de bază ale criptografiei moderne.

Confidențialitatea se referă la anumite reguli și linii directoare executate de obicei în baza acordurilor de confidențialitate care asigură că informațiile sunt limitate la anumite persoane sau locuri.

Integritatea datelor se referă la menținerea și asigurarea faptului că datele rămân exacte și consecvente pe întregul ciclu de viață.

Autentificarea este procesul prin care se asigură că data revendicată de utilizator îi aparține.

Nerepudierea se referă la capacitatea de a se asigura că o persoană sau o parte asociată cu un contract sau o comunicare nu poate nega autenticitatea semnăturii sale asupra documentului său sau a trimiterii unui mesaj.

14 Criptarea cu cheie publica si cheie private

Criptarea cu cheie publica foloseste o pereche de chei legate de matematica. Un mesaj care este criptat cu prima cheie trebuie decriptat cu a doua cheie, iar un mesaj care este criptat cu a doua cheie trebuie decriptat cu prima cheie.

Fiecare participant la un system de chei publice are o pereche de chei. O cheie este nominalizata ca cheie private si este tinuta secret. Cealalta cheie este distribuita oricui o doreste, aceasta cheie este cheia publica.

Pentru a contracara potentialul de chei falsificate, sistemele de chei publice ofera mecanisme pentru validarea cheilor publice si a altor informatii cu certificate digitale si semnaturi digitale, un exemplu de certificate poate fi vazut in fig ()

Criptare cu cheie privata foloseste o singura cheie care este partajata intre expeditor si destinatar. Ambele trebuie sa aiba cheia, expeditorul cripteaza mesajul folosind cheia, iar receptorul decripteaza mesajul cu aceeasi cheie.Atat expeditorul, cat si destinatarul trebuie sa pastreze cheia private pentru a pastra comunicarea privata.

Cateva diferente dintre cheia privata si cheia pubilica:

Cheie private

-este folosita atat pentru criptarea, cat si pentru decriptarea datelor sensibile. Este partajata intre expeditor si destinatarul dateleor criptate.

-performanta cheii private este mai rapida.

-cheia private este pastrata secret si nu publicata pentru nimeni, in afara de expeditor si destinatar.

-mecanismul cheii private se numeste simetric deoarece o singura cheie este partajata intre doua parti.

Cheia publica

-este utilizata numai in scopul criptarii datelor.

-performanta cheii publice este mai lenta

-cheia publica este libera

-mecanismul cheii publice este numit asimetric deoarece exista doua chei pentru scopuri diferite

1.5 Hashing

Hashing este procesul de amestecare a informațiilor brute în măsura în care nu le poate reproduce înapoi la forma sa originală. Preia o informație și o trece printr-o funcție care efectuează operații matematice pe textul simplu. Această funcție se numește funcție hash, iar rezultatul se numește valoare hash/digest.

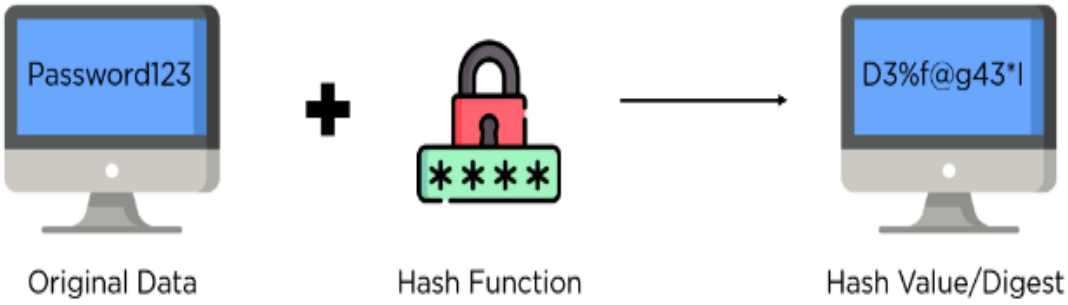


Figura.1

După cum se vede din imaginea de mai sus, funcția hash este responsabilă pentru conversia textului simplu în rezumatul hash respectiv. Ele sunt concepute pentru a fi ireversibile, ceea ce înseamnă că rezumatul dvs. nu ar trebui să vă ofere textul simplu original prin niciun mijloc necesar. Funcțiile hash oferă, de asemenea, aceeași valoare de ieșire dacă intrarea rămâne neschimbată, indiferent de numărul de iterații.

Exista doua aplicatii principale ale hashingului:

Hash-uri parole: În majoritatea serverelor de site-uri web, acesta convertește parolele utilizatorului într-o valoare hash înainte de a fi stocat pe server. Acesta compară valoarea hash recalculată în timpul conectării cu cea stocată în baza de date pentru validare.

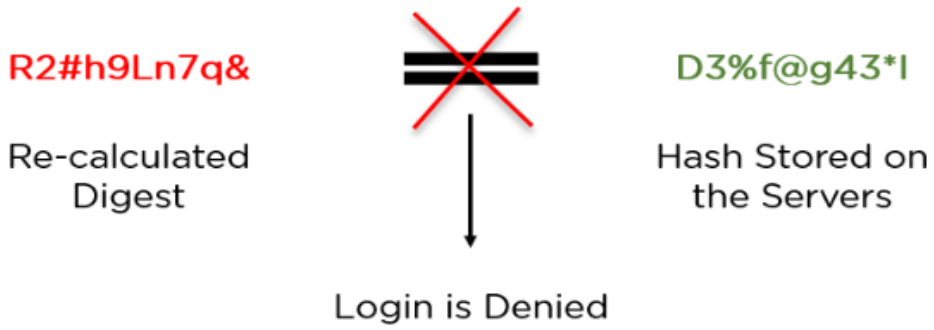


Figura.2

Verificarea integrității: atunci când încarcă un fișier pe un site web, și-a partajat hash-ul ca un pachet. Când un utilizator îl descarcă, poate recalcula hash-ul și îl poate compara pentru a stabili integritatea datelor.

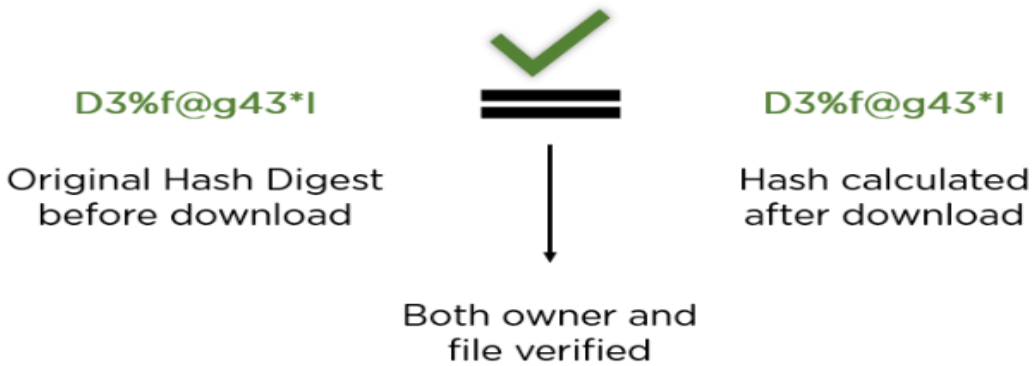


Figura.3

2.Descrierea Aplicatie

Aplicațiile de Password Manager sunt concepute pentru a fi sigure și ușor de utilizat, astfel încât să puteți stoca și gestiona cu ușurință parolele pentru conturile dvs. online. Acestea folosesc criptarea pentru a proteja informațiile stocate și, în general, necesită o singură parolă principală pentru a accesa seiful de parole. Această parolă trebuie să fie puternică și unică, iar unele aplicații pot oferi chiar și autentificare cu factori multipli (de exemplu, autentificarea cu amprenta digitală).

Atunci când utilizați o aplicație de Password Manager, puteți introduce și stoca parole puternice și unice pentru fiecare cont în parte. Aceste parole sunt generate automat de aplicație și sunt foarte greu de ghicit sau de spart. În plus, majoritatea aplicațiilor de Password Manager vă permit să stocați și alte informații importante, cum ar fi numele de utilizator, adresa de e-mail, numerele de carduri de credit și adresele de livrare.

Principalele caracteristici ale unei astfel de aplicații includ generarea de parole puternice, stocarea acestora într-un seif virtual securizat, sincronizarea între diferite dispozitive și introducerea automată a acestora atunci când sunt necesare.

În general, utilizarea unei aplicații de Password Manager vă poate ajuta să vă protejați conturile online și să vă gestionați parolele în mod eficient și sigur. Cu toate acestea, trebuie să fiți conștienți de faptul că aplicațiile de Password Manager nu sunt imune la atacuri și, prin urmare, este important să le utilizați în combinație cu alte măsuri de securitate, cum ar fi autentificarea cu doi factori și actualizarea frecventă a parolelor.

În demo-ul prezentat de către echipa noastră se poate obseva că aplicatia este scrisă in limbajul de programare dinamic Python și folosește ca funcție de hash criptografica SHA-256, un algoritm mathematic care transformă datele de intrare (mesajul) intr-o valoare de ieșire (hash) de o dimensiune fixă, care este unică și semnificativă pentru mesajul respectiv. Aplicația dispune de o bibliotecă software de bază de date numită SQLite3, care oferă o metodă ușoară de a gestiona bazele de date. O caracterisită important a SQLite3 este faptul că nu necesită un server de bază de date separate.

Interfața grafică a programului este formată cu ajutorul Tkinter, este o bibliotecă standard de interfețe grafice (GUI) pentru limbajul de programare Python. Aceasta oferă un set de widget-uri grafice, cum ar fi butoane, etuchete, casete de slectare, liste de selecție și ferestre de dialog.

3.Tutorial de folosire a aplicatiei

Flosirea aplicației este foare simplă, pentru început trebuie sa alegemo o parola formată din lietere mari, mici cat și simboluri și cifre, aceasta fiind ținuta minte de catre utilizator. Dupa adaugatea parolei se apasa butonul **save**, în interiorul aplicației se va creea o bază de date locală prin intermediul librariei SQLite3.

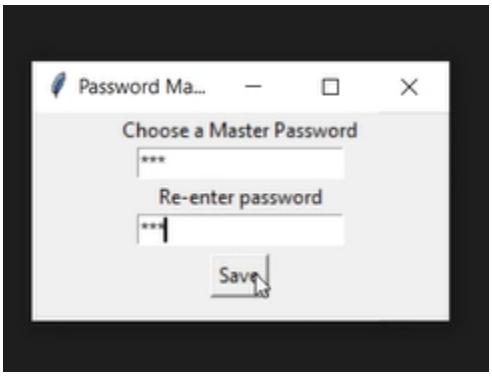


Figura.4

Dupa apasarea butonului **save** o sa ne fie generată o cheie prin limbajul de cripare SHA-256, care va putea fii folosită pentru recuperarea contului și pe care o putem copia in clipboard prin apăsarea butonului “Copy Key” (Figura 5)

Figura.5

Iar prin apasarea butonuli **done** o sa fim redirectionati catre ecranul principal al aplicatiei. (Figura 6)

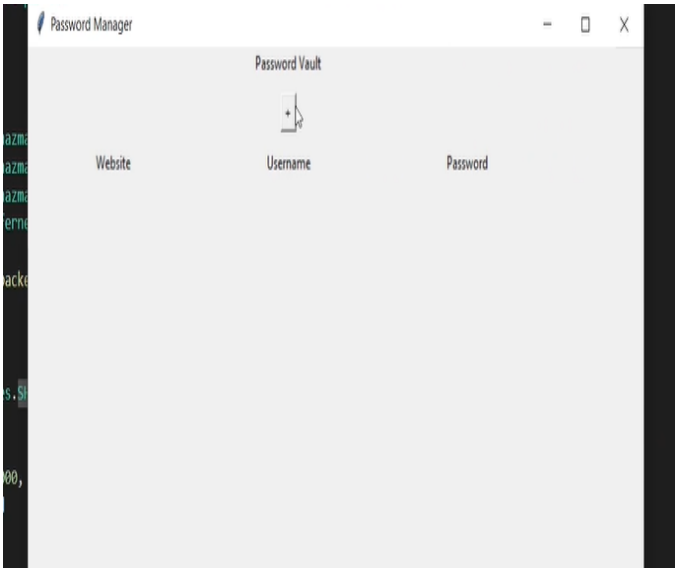


Figura.6

Prin apasarea butonului “**+**“de pe ecranul principal putem adauga numele site ului, numele de utilizator si a parola cu care contul este asimilata, avand posibilitatea și de a le șterge din componența bazei de date si a interfeței vizuale prin apăsarea butonului **DELETE** .

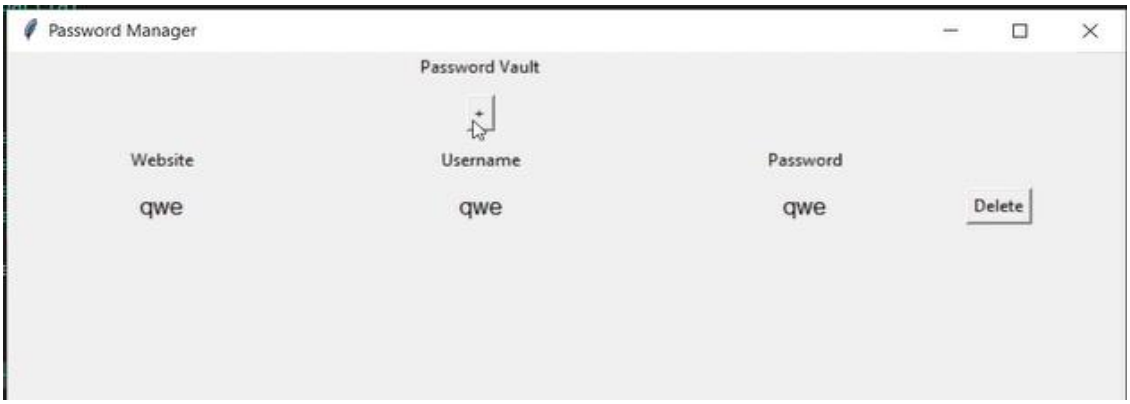


Figura.7

În cazul în care fară să vrem închidem aplicația, baza de date va fi salvată automat, iar prin repornirea aplicatiei utilizatorul trebuie doar să introducă parola pe care a ales-o atunci cand a deschis pentru prima data aplicatia și a introdus o parola initiala.

Figura.8

În cazul în care uitam parola principala, avem opțiunea de resetare a parolei care ne va trimite intr-o fereastra separata unde trebuie sa introducem parola generata prin SHA-256, aceasta in cele din urma daca este corecta ne va permite introducerea unei alte parole exact ca prima oara cand am deschis aplicatia, dejavantatul ar fi ca tot

continutul care se afla in baza de date o sa fie sters. Iar in cazul in care se pierde si parola generada prin SHA-256 tot ce ramâne de facut este să ștergem baza de date, toate datele salvate fiind pierdute.

4.Tehnologii folosite

In realizarea proiectului Password Manager am folosit baza de date sqlite 3, interfata grafica tkinter, hashlib, base64, identificatorul unic uuid, limbajul de hasurare SHA256 si Python.

Cateva notiuni informative despre SHA 256:

SHA 256 genereaza o semnatura digitala aproape unica de 256 de biti pentru un text.

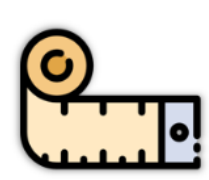
SHA 256 este una dintre functiile hash succesoare pentru SHA 1 si este una dintre cele mai puternice hash disponibile.

Unele dintre caracteristicile remarcabile la SHA sunt urmatoarele:

- Lungimea mesajului: lungimea textului clar trebuie sa fie mai mica de 264 de biti.Marimea trebuie sa fie in zona de comparative pentru a pastra digeratul cat mai aleator posibil.

- Lungimea rezumatului: lungimea rezumatului hash ar trebui sa fie de 256 de biti in algoritmul SHA 256, 512 de biti in SHA 521. Rezumatele mai mari sugereaza de obicei mult mai multe calcule cu pretul vitezei si spatiului.

- Ireversibil: prin proiectare, toate functiile hash, cum ar fi SHA 256, sunt ireversibile. Nu ar trebui sa obtinem un text simplu atunci cand digeratul in prealabil si nici nu ar trebui sa ne ofere initiala atunci cand trecem din nou prin functia hash.



Message Length

Digest Length

Irreversible

Figura.8

Python este un limbaj de programare de nivel inalt, interactive, orientat pe obiecte si de uz general foarte popular.

Python a fost creat in 1989 de Guido van Rossum fiind unul dintre cele mai populare limbaje de programare din lume.

Un avantaj major pe care Python il are este comunitatea mare si active de dezvoltare si utilizatori care contribuie cu module si biblioteci noi, aceste lucruri fac ca python sa fie cautat pentru analiza de date si machine learning, existand o varietate foar te larga.

In concluzie, Python este un limbaj puternic, flexibil si usor de invata care poate sa ruleze pe multe platforme cum ar fi: Windows, Linux, etc.

Modulul SQLite 3 a fost creat de Gerhard Haring, oferind o interfata SQL comapatibila cu specificatiile DB-API 2.0 descrisa de Pep 249 si necesita SQLite 3 sau mai nou.

SQLite3 este o bibliotecă software de baze de date, care oferă o metodă ușoară de a gestiona baze de date în cadrul aplicațiilor software. Acesta este un sistem de gestiune a bazelor de date relaționale, ceea ce înseamnă că datele sunt organizate în tabele cu rânduri și coloane. SQLite3 poate fi utilizat cu succes în aplicații mobile și desktop, pentru a stoca și gestiona date.

O caracteristică importantă a SQLite3 este faptul că nu necesită un server de bază de date separat, ceea ce o face o opțiune populară pentru dezvoltatorii de aplicații care doresc o soluție de bază de date încorporată în aplicațiile lor. SQLite3 este de asemenea foarte ușor de utilizat și integrat în proiecte software, deoarece este disponibil ca o bibliotecă software încorporabilă într-un cod sursă.

Cu SQLite3, dezvoltatorii pot crea baze de date de dimensiuni moderate și mari, care pot fi accesate și actualizate prin intermediul interfeței SQL (Structured Query Language). De asemenea, SQLite3 oferă suport pentru tranzacții și declanșatoare, care pot fi utilizate pentru a asigura integritatea datelor și pentru a implementa logica de afaceri în bazele de date.

În concluzie, SQLite3 este o bibliotecă software de bază de date relațională, care poate fi utilizată pentru a gestiona date în cadrul aplicațiilor mobile și desktop, fără a necesita un server de bază de date separat.

Tkinter este o bibliotecă standard de interfață grafică (GUI) pentru limbajul de programare Python. Aceasta oferă un set de instrumente pentru crearea de interfețe grafice utilizator pentru aplicații desktop.

Tkinter este bazată pe biblioteca grafică Tcl/Tk, care este populară pentru crearea de interfețe grafice pentru multe limbaje de programare. Tkinter oferă un set de widget-uri grafice, cum ar fi butoane, etichete, casete de selectare, casete de selectare multiple, liste de selecție și ferestre de dialog, care pot fi utilizate pentru a crea interfețe grafice complexe.

Oferind o multitudine de opțiuni pentru personalizarea aspectului și a comportamentului widget-urilor, Tkinter poate fi utilizat pentru crearea de aplicații desktop cu interfețe grafice personalizate și atractive. De asemenea, datorită integrării sale strânse cu Python, este ușor de utilizat și poate fi combinat cu alte biblioteci Python pentru a crea aplicații puternice și extensibile.

Base64 este un sistem de reprezentare a datelor binare într-o formă ușor de transmis și de stocat, care utilizează un set de 64 de caractere. Este utilizat în mod frecvent în aplicații de rețea pentru a transfera date, cum ar fi imagini, fișiere audio și video și alte tipuri de fișiere, prin intermediul protocoalelor de rețea care nu permit transferul direct de date binare.

În mod specific, Base64 transformă fiecare grup de 3 octeți (24 de biți) din datele binare într-un grup de 4 caractere, fiecare caracter reprezentând 6 biți. Caracterele utilizate în Base64 sunt de obicei litere majuscule și minuscule, cifre și caractere speciale, cum ar fi "+" și "/". De exemplu, textul "Hello, world!" poate fi reprezentat în Base64 ca "SGVsbG8sIHdvcmxkIQ==".

Base64 este util în situațiile în care trebuie să se trimită date binare prin intermediul unui protocol care nu suportă transferul direct de date binare. De exemplu, email-urile nu pot conține date binare, deci acestea trebuie să fie

convertite în format Base64 înainte de a fi trimise. De asemenea, Base64 este utilizat în aplicații web pentru a transmite imagini sau alte fișiere de către server la client, prin intermediul codării Base64 a datelor.

Hashlib este o librărie standard de Python care oferă funcționalitate de criptare hash. Aceasta permite crearea de hash-uri pentru date și verificarea integrității datelor prin compararea hash-urilor.

Hash-urile sunt valori unice generate dintr-un input de date, astfel încât același input va genera întotdeauna același hash. Un algoritm de hash criptografic puternic poate fi utilizat pentru a verifica integritatea datelor și pentru a se asigura că datele nu au fost modificate în timpul transmiterii sau stocării.

Librăria **hashlib** oferă o serie de algoritmi de hash, inclusiv MD5, SHA-1, SHA-256 și SHA-512. Pentru a genera un hash, se apelează funcția corespunzătoare algoritmului de hash, se furnizează datele de intrare și se obține valoarea hash.

UUID este prescurtarea de la "Universally Unique Identifier" (Identificator Universal Unic) și reprezintă un identificator standard utilizat în tehnologia informației. Acesta este format dintr-un șir de caractere de 36 de caractere hexazecimale, împărțit în 5 grupuri, separate prin cratime.

Un exemplu de UUID ar fi: **550e8400-e29b-41d4-a716-446655440000**.

UUID-urile sunt folosite pentru a identifica în mod unic resursele, obiectele sau entitățile dintr-un sistem. Ele sunt foarte utile în aplicații distribuite și în rețele de calculatoare, deoarece permit identificarea unică a entităților fără a fi nevoie să se coordoneze între mai mulți utilizatori sau sisteme.

Există mai multe tipuri de UUID-uri, dintre care cele mai comune sunt:

- UUID versiunea 1: generează un identificator bazat pe timp și pe adresa MAC a dispozitivului
- UUID versiunea 4: generează un identificator aleatoriu, bazat pe numere aleatorii generate de către sistemul de operare

OS este o bibliotecă standard din Python care oferă o interfață pentru interacțiunea cu sistemul de operare (Operating System). Aceasta permite programatorilor să creeze programe care pot accesa, gestiona și utiliza resursele de sistem, cum ar fi fișierele, directoarele, procesele și variabilele de mediu.

Biblioteca **OS** include o serie de funcții și metode care permit programatorilor să efectueze o varietate de acțiuni, cum ar fi:

- Accesarea și manipularea fișierelor și directoarelor
- Interacțiunea cu variabilele de mediu
- Controlul proceselor și al fluxurilor de date
- Lucrul cu calea sistemului de operare
- Interacțiunea cu utilizatorii sistemului prin intermediul terminalului

Biblioteca **OS** este disponibilă pe majoritatea platformelor de sistem de operare, inclusiv Windows, Linux și macOS. Această bibliotecă este foarte utilă pentru dezvoltarea de programe care trebuie să comunice cu sistemul de operare pentru a efectua acțiuni specifice și pentru a accesa resursele de sistem.

Pyperclip este o bibliotecă Python care oferă funcționalitatea de copiere și lipire în clipboard-ul sistemului de operare. Această bibliotecă poate fi utilizată pentru a copia text în clipboard-ul sistemului de operare sau pentru a prelua text din clipboard și a-l utiliza în programele Python.

Pyperclip este o bibliotecă ușor de utilizat și suportă diverse tipuri de date, inclusiv text și imagini. Pentru a utiliza **pyperclip**, trebuie să instalați biblioteca înainte de a o putea importa în programul dvs. Python

Functools este o bibliotecă standard din Python care oferă funcții utile pentru programarea funcțională. Această bibliotecă include funcții care ajută la crearea și manipularea funcțiilor, precum și funcții pentru gestionarea și utilizarea obiectelor **functor**.

Iată câteva dintre funcțiile utile oferite de biblioteca **functools**:

- **partial()** - permite crearea de funcții noi prin fixarea unuia sau mai multor argumente ai unei funcții existente.
- **reduce()** - aplică o funcție pe toate elementele unei secvențe și returnează un singur rezultat.
- **wraps()** - permite decorarea unei funcții pentru a păstra atributele originale ale acesteia, cum ar fi numele sau documentația.

5.Testare

Testarea a reprezentat o parte importanta a procesului de dezvoltare a aplicației. Scopul testelor a fost de a verifica funcționarea corectă a tuturor componentelor aplicației , de la interfața grafică a aplicației, criptarea cu SHA-256 și pană la baza de date formată din SQLite3.

Testele au fost realizate în mai multe etape . În primul rând s-au realizat teste unitare pentru fiecare componenta a aplicației cu scopul verificării funcționalității corecte a acestora. Au fost utilizate biblioteci diferite pentru fiecare componentă a aplicatiei, precum uuid, base64, hashlib, tkinter.

Pe parcurs am întâmpinat mai mai multe probleme de dezvoltare care au necesitat soluționare imediată pentru a putea continua cu proiectul. Pentru a face față acestor provocări, am urmat următorii pași:

1. Identificarea problemei: Înainte de a încerca să rezolvăm o problemă, am petrecut timpul necesar pentru a înțelege clar problema și pentru a identifica factorii care o cauzează. Această etapă ne-a permis să găsim soluții potențiale și să evităm presupunerile greșite.

2. Testarea și validarea soluțiilor: După identificarea soluțiilor potențiale, am testat fiecare soluție într-un mediu de dezvoltare separat, utilizând metode de testare adecvate. Acest lucru ne-a permis să validăm eficacitatea fiecărei soluții și să selectăm soluția cea mai bună.

3. Implementarea și testarea finală: După ce am identificat și validat soluția optimă, am implementat soluția și am testat-o în mediul de dezvoltare final. Am asigurat că soluția implementată este scalabilă, flexibilă și îndeplinește cerințele și specificațiile proiectului.

4. Îmbunătățirea continuă: După implementarea soluției și finalizarea testelor finale, am continuat să îmbunătățim codul și să căutăm metode de optimizare a soluției noastre. Această abordare ne-a permis să îmbunătățim în mod constant soluția și să îndeplinim cerințele.

Pentru a asigura calitatea proiectului , s-au realizat și teste de performanță pentru a verifica timpul de răspuns al aplicației și consumul de resurse ale calculatorului. Aceste teste au fost realizate utilizând diferite metode introducere a parolelor in moduri greșite și incorecte și au demonstrat faptul că aplicația funcționează eficient și rapid.

În concluzie , testarea a reprezentat o parte importantă a dezvoltării aplicației Password Manager și ne-am asigurat că aplicația funcționează corespunzător și eficient. Testele de integrare și performanță au demonstrat că aplicația este capabilă să recunoască caracterele introduse greșit, aceasta revocând accesul persoanelor cu intenții rele în a intra în aplicațiia propriu-zisă și sa ne asigure ca aplicația este securizată de catre orice atac cibernetic.

6.Management

În timpul implementării proiectului , s-au întâmpinat anumite provocări de management care au necesitat o abordare atentă și soluționarea lor în mod eficient pentru a se asigura succesul proiectului. Unele dintre lecțiile importante învățate în timpul managementului proiectului includ :

1. Planificarea și organizarea – A fost important să se planifice și să organizeze fiecare etapă a proiectului în mod atent, inclusive definirea obiectivelor, stabilirea termenelor limită, alocarea resurselor , precum și definirea sarcinilor și responsabilităților . O bună planificare și organizare au contribuit semnificativ la atingerea obiectivelor proiectului în timp și cu resurse minime.

2. Comunicarea eficientă – Comunicarea eficientă între membrii echipei a fost crucială în reușita proiectului. A fost important să se stabilească metode de comunicare clare și să se comunice constant cu toți membrii echipei pentru a se asigura că toată lumea este pe aceeași lungime de undă și se lucrează la obiectivele comune.

3. Managementul riscului – Identificarea și gestionarea riscurilor proiectului a fost o parte important a procesului de management . Au fost luate în considerare riscurile posibile și s-au luat măsuri preventive pentru a minimiza impactul acestora în cazul apariției lor.

4. Evaluarea continuă - A fost important să se efectueze evaluări continue ale proiectului pentru a identifica problemele și pentru a implementa soluții imediate. Aceste evaluări m-au ajutat atât pe mine ca manager cât și pe membrii echipei și să ajustăm strategia și planurile pentru a îndeplini obiectivele proiectului.

Am folosit Jira ca o platformă de management al proiectelor dezvoltată de compania Atlassian. Este utilizată de organizații din întreaga lume pentru a gestiona sarcinile, problemele, proiectele și rapoartele. Jira oferă un mediu centralizat în care echipele pot colabora, urmări progresele și lua decizii informate pe baza datelor și analizelor în timp real.

Funcționalitățile cheie ale Jira includ: managementul proiectelor și al sarcinilor, urmărirea problemelor și a bug-urilor, managementul timpului și al costurilor, raportarea și analiza datelor, colaborarea și comunicarea în cadrul

echipei. Utilizatorii pot personaliza fluxurile de lucru, tablourile Kanban și rapoartele pentru a se potrivi nevoilor lor specifice de afaceri.

Jira oferă integrări cu alte instrumente și platforme de dezvoltare, cum ar fi GitHub, Bitbucket și Confluence, pentru a facilita colaborarea și managementul într-un mod mai eficient. În plus, Jira oferă o gamă largă de aplicații și plug-in-uri pentru a adăuga funcționalități suplimentare și pentru a îmbunătăți performanța.

În general, Jira este o platformă puternică și scalabilă pentru managementul proiectelor, utilizată de companii de toate dimensiunile și din toate industriile.

Despre metodologia de dezvoltare , am ales să folosim o abordare Agile pentru a putea gestiona mai eficient proiectul și pentru a putea face față oricăror schimbări sau cerințe care apar pe parcurs , de aceea am ales să

folosim platforma Jira pentru a ne organiza backlog-ul , pentru a urmări progresul sarcinilor și a prioritiza activitățile . Am împărțit proiectul în sprint-uri , astfel încât să putem avea un obiectiv clar pentru fiecare săptămână și să ne asigurăm că progresăm în mod eficient .

Am ținut legătura in fiecare zi cu membrul echipei , pentru a monitoriza progresul și face modificări , ajustări acolo unde a fost necesar.

Prin utilizarea metodei Agile , am reușit să livrăm o aplicație funcțională într-un timp destul de scurt și cu o echipă restrânsă.

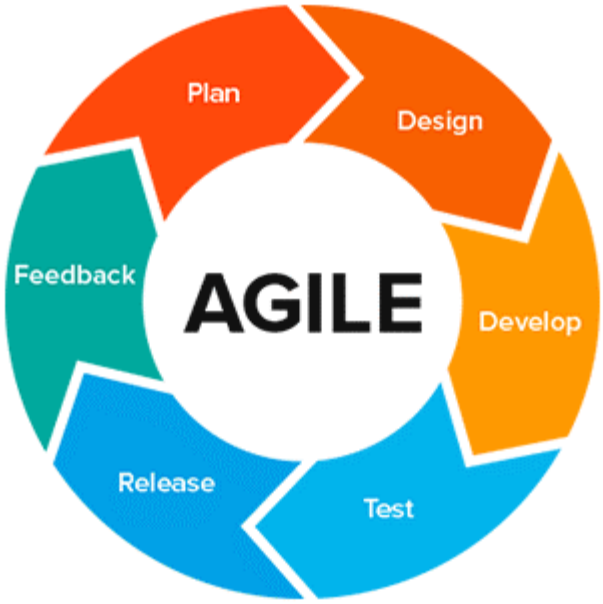


Figura.9

7. GITHUB REPOSITORY

https://github.com/AdryanT/Proiect-ISW -> Link GitHub

Repository GitHub care cuprinde codul sursa al aplicatiei create Password Manager.

Folosirea repository pe Github a cuprins o parte importanta in realizarea proiectului, oferind un mediu ostil si bine pus la punct pentru gestionarea codului intre toti membrii echipei.

La inceput, a fost creeat un nou repository, configuranduse si setarile de baza. Mai apoi adaugarea tuturor membrilor echipei pentru a putea contribui la realizarea si dezvoltarea proiectului.

Fisierul principal care contine codul sursa al aplicatiei fiind password_.py

Pe parcursul avansarii in dezvoltarea proiectului am intretinut si actualizat repository-ul, asigurandune ca mereu este actualizat si in pas cu noile modificari.

8.Lectii invatate

Desfășurarea acestui proiect a implicat provocări și obstacole dar și o serie de lecții învățate care ne-au ajutat să ne îmbunătățim abilitățile și să facem față cu succes altor proiecte în viitor.

În cele ce urmează , voi enumera cele mai importante lecții pe care le-am învățat în timpul dezvoltării acestui proiect:

1. Planificarea este cheia succesului. Am învățat că planificarea este esențială pentru a ne asigura că toate sarcinile sunt realizate în timp util și că echipa lucrează într-un mod coerent și organizat. Prin crearea unui plan de proiect clar și detaliat, am reușit să evităm confuzia și să ne asigurăm că toate sarcinile au fost realizate într-un mod eficient.
2. Comunicarea este esențială. Am învățat că comunicarea deschisă și eficientă este esențială pentru succesul unui proiect. Am avut întâlniri regulate în care am discutat despre progresul proiectului, problemele întâmpinate și soluțiile găsite. Prin aceasta am reușit să ne menținem la unison și să ne asigurăm că toți membrii echipei știu exact ce trebuie să facă și când.
3. Învățarea continuă este importantă. Am învățat că este important să fim deschiși la noi idei și să ne îmbunătățim constant abilitățile. În timpul dezvoltării proiectului, am descoperit noi tehnologii și abordări pe care le-am integrat în proiect. De asemenea, am participat la cursuri și am citit documentații pentru a ne îmbunătăți cunoștințele.
4. Gestionarea timpului este cheia. Am învățat că gestionarea timpului este una dintre cele mai importante aspecte ale dezvoltării unui proiect. Prin planificarea eficientă și distribuirea sarcinilor în mod adecvat, am reușit să ne îndeplinim obiectivele în timp util.
5. Flexibilitatea este importantă. Am învățat că trebuie să fim flexibili și să ne adaptăm la schimbările care apar în timpul dezvoltării proiectului. Uneori, unele sarcini pot fi mai dificile decât altele, sau pot fi necesare modificări ale planului de proiect. În aceste situații, trebuie să fim deschiși la schimbare și să ne adaptăm pentru a atinge obiectivele noastre.
6. Încrederea în echipă este crucială. Am învățat că încrederea în membrii echipei este crucială pentru a realiza cu succes un proiect. Prin asigurarea că fiecare membru al echipei are o sarcină clară și este responsabil pentru realizarea acesteia, am putut să ne bazăm unii pe a

9.Concluzii

După finalizarea acestui proiect, am ajuns la următoarele concluzii:

- Criparea este un domeniu complex si interesant, cu o varietate de aplicatii practice și un potential enorm de dezvoltare
- Implementarea unui Password Manager este o provocare dar și o oportunitate de a învața și de a experimenta cu tehnologii noi.
- Utilizarea algoritmilor de hasurare sunt o soluție eficienta pentru protejarea parolelor de atacurile cibernetice.
- Integrarea bibliotecii tkinter si SQLite3 in proiectul nostru a permis integrarea mult mai usor a unei interfețe prietenoase și a unei baze de date ușor de creeat.
- În timpul dezvoltarii proiectului, am învătat sa lucram cu tehnologii si instrumente noi precum, tkinter, haslib, SQL , GITHUB si altele.
- Am învățat ca documentarea și planificarea sunt esențiale pentru un proiect de success și ca comunicarea eficientă între membrii echipei este cheia pentru a Evita erorile și pentru a accelera dezvoltarea.

Desfășurearea proiectului Password Manager a fost o experiență și provocare, dar și extrem de satisfăcătoare pentru mine și echipa mea. În urma acestui proiect, am învățat multe lucruri noi și ne-am dezvoltat abilitățile de programare, de management de proiect și de colaborare în echipă.

Am avut parte de provocări și de momente de frustrare, dar am învățat să le gestionăm eficient și să găsim soluții în timp util. Colaborarea în echipă a fost foarte importantă pentru reușita proiectului și am învățat să lucrăm împreună pentru a atinge obiectivele setate.

De asemenea, am învățat să lucrăm cu tehnologii noi și să le integrăm în proiectul nostru, cum ar fi biblioteca SHA-256, SQLite3, tkinter. Acest lucru ne-a adus o experiență valoroasă și ne-a îmbunătățit cunoștințele în domeniul programării.

Un alt aspect important pe care l-am învățat în cadrul acestui proiect este importanța testării și a documentației. Am învățat să testăm eficient codul nostru și să scriem o documentație clară și cuprinzătoare pentru a ne ajuta colegii și utilizatorii să înțeleagă mai bine proiectul și să lucreze cu el.

În concluzie, dezvoltarea proiectului OCR Scanner ESP32-CAM a fost o experiență foarte valoroasă și am învățat multe lucruri noi, atât din punct de vedere tehnic, cât și din punct de vedere al managementului de proiect și al colaborării în echipă. Sperăm ca proiectul nostru să fie util pentru alți utilizatori și să aducă o contribuție semnificativă în domeniul tehnologiei.

BIBLIOGRAFIE

https://www.techopedia.com/definition/31435/password-manager

https://moodle.usm.md/mod/assign/view.php?id=57528

https://www.academia.edu/43108459/CURS_TEHNICI_DE_SECURIZARE_A_INFORMA%C8%9AIEI_TEHNICI_DE_SECURIZARE_%C8%98I_CRIPTARE

https://www.trustedreviews.com/best/best-password-manager-4019795

https://timr.com.br/entenda-a-criptografia-e-seus-beneficios/

https://www.simplilearn.com/tutorials/jira/what-is-jira-and-how-to-use-jira-testing-software

https://corporatefinanceinstitute.com/resources/management/management-skills/

https://www.bestjobs.eu/casual/2022/06/02/ce-este-metodologia-agile-si-ce-avantaje-are-intr-o-companie/