**Mini Project Report on**



**Password Manager**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name : Gaurang Sharma University Roll No. 2018801**

***Under the Mentorship of***

**Ms. Tanusha Mittal**

**Assistant professor department of CSE**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

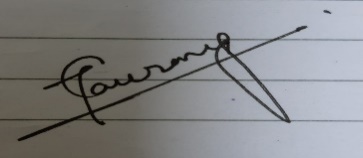
**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Password manager”** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms. Tanusha Mittal, assistant professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name : Gaurang Sharma    University Roll no. 2018801

Signature : 

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction |  |
| Chapter 2 | Literature Survey |  |
| Chapter 3 | Methodology |  |
| Chapter 4 | Result and Discussion |  |
| Chapter 5 | Conclusion and Future Work |  |
|  | References |  |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

1. **Introduction**

We will be creating an application which helps to take notes at any time and helps to generate a very strong password for the security of your account and makes them available whenever you need. It's going to have two major functionalities at the same time.

Password managing & note-taking are one of the most overlooked virtual activities in today's digital age. Various companies are constantly adding new features to note-taking and password management. So, instead of having and maintaining two different apps in your system, we'll be developing a single app which can perform both of these activities.

This project is a simple application that allows users to store and manage their notes and passwords in a secure way. The application is built using Python and utilizes various libraries such as cryptography and specific recipes like fernet from it.

**Chapter 2**

**Literature Survey**

A password manager is a software dedicated to saving and managing  
personal information, usually passwords.  
Although password managers offer great benefits like faster au-  
thentication, secured storage for our personal information, and pre-  
vention against phishing and brute force attacks, their use has not  
been adopted by many users . According to a study in a large cor-  
poration, only 1% out of 836 employees that filled out a questionnaire  
stated they use some password manager .  
The investigation by Fagan et. shows that while users of pass-  
word managers noted their usefulness and convenience, others noted  
security issues as the main factor for not using password managers.  
Furthermore, the analysis of differences in emotions between “users” and “non-users” reveals that participants who use a password manager are more likely to feel secure, admiring and energetic, and less likely to feel suspicious when using their password manager to log into a website.  
We can divide password managers into multiple categories based  
on their implementation and provided functionality . An ideal  
password manager should include:  
∙ Securely stored personal information – The personal informa-  
tion written or inserted into a password manager should be  
stored in an encrypted form. The most common practice is to  
protect the vault of the password manager with a master key.  
The master key is essentially a password to enter the vault of  
passwords to keep the vault secure. Alternatively, a token can  
be used for the same purpose.  
∙ Password Generator – Users can generate stronger and safer  
passwords via password meter or generator build inside a par-  
ticular password manager. The study of web password habbits  
by Florencio et. states that an average user has pass-  
words, shared across different services and thus should have  
a unique password for each service. It ensures that if one account  
gets breached, other accounts will not be compromised.  
∙ Faster authentication – Password managers offer various func-  
tionality tools including autofilling passwords. The manager fills in the information automatically without the involvement  
of the user. It can even execute login protocols and visit pre-set  
domains which prevents phishing attacks.  
∙ Synchronization – Users typically share passwords across mul-  
tiple devices, so it is convenient to have shared access to the  
database. The synchronization can be managed automatically  
or manually.  
2.1 Usability of the Ideal Password Manager  
An essential part of the ideal password manager is usability and in-  
cluded features. To compare three such applications to an ideal man-  
ager, we define its features, describe the ideal scenario, and test each  
manager’s implementation. Then we try to choose the best approach  
by the purpose of use.

2.2 Database  
One of the essential abilities of a Password Manager is creating, maintaining, importing, exporting, and deleting databases. The database is the most important entity of a password manager and thus should be as secure as possible. Not only when the application is closed, and the database is encrypted, but even when the application is running to prevent other processes from accessing the data. Tasks that we will  
perform on our managers are:  
1. Create – As almost every manager works with a database, it  
should have an ability to create a database, ideally, at the first  
launch.

2. Import and Export – If the password manager is not cloud-  
based, it is convenient to have the ability to import and export  
the database in multiple formats to transfer the information to  
another device or manager. This type of operation should be  
secured. If not, we should delete it immediately after we are  
done using it.  
3. Management – It should be straight-forward so the user can  
modify its properties with ease. It includes creating and manag-  
ing multiple entry types and organizing them into folders for  
better readability.  
4. Synchronization – For cloud-based managers, it involves syn-  
chronizing the local database with the cloud database. Other  
types of managers should dispose of this feature on some level.

The above information was discussed in the research work of Daniel Pecuch, bachelor’s of informatica at Masaryk University.

References : https://is.muni.cz/th/sm620/pecuch\_bc.pdf

**Chapter 3**

**Methodology**

The basic approach we must follow to solve our problem statement is as follows:

Every password manager must save the password a user gives and output it when the user needs it. We are gonna apply the same thing to our password manager. We will first start by creating a .txt file to store the passwords.

## 1| Create a text file in python

Making a text file in python is fairly simple.

import os.path

def checkExistence():

if os.path.exists("info.txt"):

pass

else:

file = open("info.txt", 'w')

file.close()

At first we are checking if the file info.txt exists in our directory where the python file is. If not, just simply create one. 'w' means we are creating the file to write something.

## 2| Write to file

Alike you print in the terminal, you can similarly write into a file. just use the write to write inside a file. Here is how.

def appendNew():

# This function will append new password in the txt file

file = open("info.txt", 'a')

print()

print()

userName = input("Please enter the user name: ")

password = input("Please enter the password here: ")

website = input("Please enter the website address here: ")

print()

print()

usrnm = "UserName: " + userName + "\n"

pwd = "Password: " + password + "\n"

web = "Website: " + website + "\n"

file.write("---------------------------------\n")

file.write(usrnm)

file.write(pwd)

file.write(web)

file.write("---------------------------------\n")

file.write("\n")

file.close

At first, we are opening the file we just have created. 'a' argument means we will append something in that file. We could have used 'w' which stands for write. But every time we open a file with a 'w' argument, it erases everything written previously. Which we don't want at all. So, we will continue with 'a'.

Then we take input from the user about his user name, password and the website. I'm using the empty print statement to space things out in the terminal so that it looks good. Then we are simply creating three string variables to store the username, password and website.

And then we will write to our file by using the write function. Remember, unlike print, write doesn't add a new line every time we call it. So, use \n if you want to add a new line in your file. That's it! Now our user can save passwords in info.txt file(or whatever you call).

## 3| Output the password

This time we will see what passwords the user has saved. The bellow function will get the job done.

def readPasswords():

file = open('info.txt', 'r')

content = file.read()

file.close()

print(content)

Just like our previous function, we are opening our file at first. But this time, instead of append, we will open the file as read(use r). Then we will create a new variable content which will be the place holder of the contents in the file. And then just simply print it out. BOOM!!!

Yeah! I know this function will print all the passwords which we don't want. For that, we will make need to a search operation. But in order to keep the post as simple as possible, I will end it just right here. In the next post, we will make a search option.

**Chapter 4**

**Result and Discussion**

If you are like me, you probably have logged in many websites so far. But rarely remember any password of them. And sometimes, the websites won't let you sign up until you add a 'super-strong password.' But as a matter of sorrow, you forgot that password the next day.

There is a simple solution to this gigantic problem. Use a password manager. But nowadays, maintaining privacy in this vast online world is quite impossible unless you don't use the internet at all. But you aren't going to that, are you?

You even may find your password in the dark web selling for a couple of dollars. So, using a third-party password manager doesn't sound safe to me. You should also be concerned about this. (I am not demotivating you guys not to use a third-party password manager.)

This is a Software developed to generate a strong and secured password for a user's and save their online details into a directory that can be access anytime.

In the world where everything is changing rapidly, everybody needs technology to solve complex issues and want to keep at it with ease.

Hackers are out there ripping people off their by sneaking into their online data. That's ehy it's very crucial to safeguard your data online. Very necessary.

I have created this password manager to solve the oulined problems:

=======

1-- Strong and secure password generator 2-- Creating a directory where your online data accessibility are stored no matter how huge they are. This will save you the stress of having to memorize the password generated 3-- Ability to search from the directory for any saved website and the password you used to signed up on them.

**Chapter 5**

**Conclusion and Future Work**

## The future of passwords

What is the future of passwords? Well, first of all, we should be asking: What are the futures of passwords? Indeed, even the definition of "password" is going to be complicated in the future as every site will mean something different.

Some sites will use two-factor authentication ([2FA](https://teampassword.com/blog/password-manager-with-2fa)), while others will use a [passwordless](about:blank) system. Some devices will be dealing with [biometrics](about:blank), while others passphrases. Some users will prefer [single-sign-on (SSO)](about:blank), while others will stick to the proven security and convenience of a [password manager](https://teampassword.com/blog/whats-the-best-small-business-password-manager).

Below, we will describe just a few of the many advances that you will see as part of the future of passwords:

* Two-factor authentication
* Passphrases
* Single sign-on
* Passwordless
* Zero login
* Password managers

Don't let your company fall victim to extortion emails, credential stuffing, and other password vulnerabilities.

‏‏‎ ‎‏‏‎ ‎‏‏‎ ‎

## Two-factor authentication (2FA)

Two-factor authentication is basically what it sounds like—instead of using a single authentication step involving your username and password, there is a second step where you provide further credentials.

The most common system for 2FA includes an SMS message to your phone number. Once you type in your username and password, the system sends you a text message with a one-time password (OTP), usually a six-digit number code, to type in during the second authentication step.

This has mostly replaced an older version of 2FA where you provided questions and answers to prove you are who you say you are because they were vulnerable to social engineering. That is, malicious actors could figure out your first pet’s name, your childhood best friend, or the street you grew up on and then waltz through this second step of authentication.

While 2FA with SMS codes has helped reduce the risks of brute force or dictionary attacks, a new attack arose to break through the SMS step. Indeed, hackers have found ways to port phone numbers to a different SIM card without the owner’s permission to harvest these 2FA codes.

That is why secure 2FA OTPs are now sent to authentication apps, the most popular being Microsoft’s app called Authenticator.

‏‏‎ ‎

## Passphrases

Passphrases are, in essence, many long passwords. Instead of writing the password “dog,” you pick the phrase “I walk my dog Sparky,” which might be written without spaces as “iwalkmydogsparky.” The sheer length of the passphrase makes it more secure than the base password while not being harder to remember.

In the future of passwords, allowing for longer password lengths to accommodate passphrases is a very simple-to-implement and low-cost step in higher security. Unfortunately, many sites still limit the length of passwords to 12 characters.

Passphrases can also be adapted similarly to normal passwords by changing the capitalization of different letters, adding numbers and characters, etc. For example, “iwalkmydogsparky” can then be changed to “1.w4lk-My\_d0g=Sp4rKy”.

The benefit of passphrases is that, even though they are very long and hard to guess, they are often quite easy to remember compared to a random set of letters, numbers, and characters.

‏‏‎ ‎

## Single sign-on (SSO)

The single-sign-on system allows users to use a single set of credentials once to use many separate services. This is done by all the services having trust relationships. As you move from one secure site to another, the [SSO](about:blank) system passes tokens behind the scenes to confirm you have permission to enter the site.

‏‏‎ ‎

## Passwordless

Passwords have gone from the entirety of digital security to its greatest liability. This has led many people to argue the future of passwords should be no passwords at all. [Biometrics](about:blank) has been at the forefront of the [passwordless](about:blank) revolution. Fingerprints and/or facial recognition are increasingly used by portable devices such as smartphones.

It is likely that passwordless solutions will be a part of any login ecosystem of the future. Whether using SSO or 2FA, biometrics are likely to be part of the future of passwords.

‏‏‎ ‎

## Zero Login

Some companies might do away with logins entirely. While your bank might actually call you when you try to log in from a new IP, Amazon is currently testing systems that will measure your typing speed and pressure to constantly confirm your identity while you use their site.

In these systems, the goal is to confirm your identity constantly through behavioral means and then only ask a password when the system detects anomalies in how you type, search, etc. In this case, only hackers will be prompted for passwords.

‏‏‎ ‎

## Password managers

While the future of passwords is diverse systems designed to keep you more secure while not taxing your brain remembering a longer and longer list of more and more complex passwords, passwords will not disappear entirely.

These complex passwords—and a new one for every site—are paramount to your security. For now, the best solution is a very secure and user-friendly password manager that takes all the difficulty out of using passwords.

The best way to secure your IT infrastructure is to use a password manager that includes sharing features. A password manager like TeamPassword offers high-level encryption and two-factor authentication so that only the right people can make sense of the passwords.

Before anyone can access the list of shared passwords, they must log in to the platform using their personal password and a short-term authentication code.

**References :**

▪Python Crash Course: A Hands-On, Project-Based Introduction to

Programming (2nd Edition)

▪Python Programming: An Introduction to Computer Science (3rd Edition)

❖Websites:

https://www.python.org/

https://www.geeksforgeeks.org/

https://www.wikipedia.org/

https://github.com/

https://teampassword.com/blog/what-is-the-future-of-passwords