MyPass- Password Manager

—

Name: Anit Mangal

Class: XII-F
Roll no: 5

School: DLF Public School

Session: 2020-21

Index

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Ms. Ansara Banu as well as my principal Ms. Seema Jerath who gave me the golden opportunity to do this wonderful project on the topic “Computational Programming and Thinking with Database Management”, which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Name: Anit Mangal

Class: XII-F

Certificate

This is to certify that Anit Mangal of class 12 has successfully completed his project work of Computer Science for class XII practical examination of the Central Board of Secondary Education in the year 2020-2021. It is further certified that this project is the individual work of the candidate.

Signature:

Requirements

The following programs are required before running the project:-

1. Python 3.7 or above
2. MySQL 8.0 or above

Overview

A password manager is a computer program that allows users to store, generate, and manage their personal passwords for online services. A password manager assists in generating and retrieving complex passwords, potentially storing such passwords in an encrypted database or calculating them on demand.

Goals

1. All basic functions, Storing, Retrieving, and Securing should be as efficient and effective as possible.
2. The password should be encrypted and stored in a safe vault.

Specifications

MyPass is a personal password manager available offline that offers an easy way to save passwords in an encrypted form for different sites in a very organized manner. To create MyPass, we've integrated Python with MySQL as a database management system.

Purpose

For many of us, working from home has become the new normal and that means securing our online accounts with strong passwords is more important than ever. But memorizing dozens and dozens of passwords can be a challenge, and using the same old password over and over is dangerous.

If you find yourself constantly getting locked out of one online account or another because you're drawing a blank when you try to log in, it's time to consider a password manager to help you seamlessly oversee and handle all your login credentials.

Functioning

MyPass is an offline password manager where we can make a new user for us which will be password protected in which we can save our passwords. MyPass gives us various options for all the possible functionalities related to a password manager. MyPass gives us an option to add as many sites as we want and save their usernames and passwords.

These saved credentials get encrypted and saved in a MySQL table, which is created separately for each user. MyPass also gives an option to modify these credentials or even delete them. Another unique thing about MyPass is that we can create as many users as we want for different people.

Future of MyPass

Although MyPass is an almost perfect program, a few little updates can be brought up to make the experience seamless. Future versions of MyPass may see a password reset feature with secret questions for logging into the program. Other than that, MyPass can be made a little more pleasing to the eye of the user and even more organized by using tables or some other visual changes, so a User Interface will be brought about for the same.

Development

Our Experience

We always wanted to make our own password manager as there are chances of getting viruses or getting passwords leaked by using 3rd party software. Also, most of the secure third party password managers require a paid subscription. We started out by designing a plan of how the code would run and all the functions an ideal password manager should have.

We decided to make Python the front end software by which the user would interact with the program. We used MySQL, the database management software for saving the encrypted data. We called the program “MyPass” as we wanted the user to get a feeling of ownership of their data and safety.

Learning Phase

We faced some difficulties while making the program since we wanted to give the best possible experience to the user. We overcame them by doing some needed research on some open source sites and took some help from our subject teacher.

We tried to perform encryption algorithms by ourselves by using some different ways , but it proved to be more complicated to make a random pattern every time than we anticipated. To tackle that, we used the “Fernet” function from a module called “Cryptography”.

We had also realised that there could be many areas where the program could give errors. We tested almost all of them and tried to provide a suitable error message according to the situation.

Source Code

Function definitions

```
def init(datauser, datapass):
    global data
    global cursor
    data = connector.connect(host="localhost", user=datauser,
passwd=datapass)
    cursor = data.cursor()
    try:
        cursor.execute("CREATE DATABASE MyPass")
    except:
        cursor.execute("USE MyPass")
upass = "ok"
uname = "Admin"
app = "app 1"
appuser = "usr"
with open("data.txt", "a"):
    Pass
```

#This function creates database objects for the program and accesses the MySQL database

```
def crtable(uname):
    try:
        cursor.execute('CREATE TABLE {} (id integer(2) PRIMARY KEY
AUTO_INCREMENT, App varchar(20) UNIQUE, AppUser varchar(20), keyg
varchar(50), Password Text)'.format(uname) data.commit()
    except:
        pass
```

#A table is created where all the data will be saved in the future in the database management system.

```
def create(uname, upass):
    with open("data.txt", "a+") as fl:
        fl.write("{}, {} \n".format(uname, upass))
        data.commit()
        print("You are now a registered user. ")
```

#This function makes a new user for us to save our passwords and usernames in it.

```
def fkey(uname, app):
    key = Fernet.generate_key()
    cursor.execute('INSERT INTO {}(keyg, App) VALUES ("{}",
“{}")'.format(uname, key.decode("ascii"), app))
    data.commit()
```

#The function generates a unique key for the application added

```
def fenc(psswrd, app, uname, switch, appuser, appold = None, appuserold
= None):
    if switch == True:
        fkey(uname, app)
        cursor.execute('SELECT keyg FROM {} WHERE App =
"{}"'.format(uname, app))
    else:
        cursor.execute('SELECT keyg FROM {} WHERE App =
"{}"'.format(uname, appold))
    key = cursor.fetchone()[0]
    f = Fernet(key.encode())
    enc = f.encrypt(psswrd.encode())
    denc = enc.decode('ascii')
    if switch==True:
        cursor.execute('UPDATE {} SET Password = "{}", AppUser = "{}" WHERE
App = "{}"'.format(uname, denc, appuser, app))
        data.commit()
        print("Password entered safely. ")
    else:
        return denc
```

#This function will be used to register an application and add a password for it to be saved after encryption in the database.

```
def passstr(password):
    while True:
        if (len(password)<8):
            return False
        elif not re.search("[a-z]", password):
            return False
        elif not re.search("[A-Z]", password):
            return False
        elif not re.search("[0-9]", password):
            return False
        elif not re.search("[_@$*]", password):
            return False
        elif re.search("\s", password):
            return False
        else:
            return True
```

#This function makes sure that the password for logging into MyPass is a strong password and hence it must satisfy some conditions.

```
def fdec(app, uname):
    try:
        cursor.execute('SELECT keyg FROM {} where App =
"{}"'.format(uname, app))
    except:
        print("You don't have this application registered to your username")
    key = cursor.fetchone()[0]
    dkey = key.encode()
    f = Fernet(dkey)
    cursor.execute('SELECT Password FROM {} WHERE App =
"{}"'.format(uname, app))
    pss = cursor.fetchone()[0].encode()
    dpss = f.decrypt(pss)
    cursor.execute('SELECT AppUser FROM {} WHERE App =
"{}"'.format(uname,app))
    appuser = cursor.fetchone()[0]
    print("\n \n \nApp Username is: ")
    print(appuser)
    print("\n \n \nPassword is: ")
    print(dpss.decode())
    print("\n")
```

#This function is used to access the stored passwords for different apps registered in it. The function decrypts and then provides us the username and passwords.

```
def mod(uname, app, appnew, psswrd, psswrdnew, appuser,appusernew)
:
    try:
        denc = fenc(psswrdnew, appnew, uname, False, appusernew, app,
appuser)
        cursor.execute('UPDATE {} SET App = "{}", Password = "{}", AppUser =
"{}" WHERE App = "{}"'.format(uname, appnew, denc, appuser, app))
        print("Modified successfully")
    except:
        print("Invalid app name")
    data.commit()
```

#The function is used to modify the application name, or the password, which will also be encrypted.

```
def deluser(uname):
    cursor.execute('DROP TABLE IF EXISTS {}'.format(uname))
    data.commit()
def delapp(uname, app):
    cursor.execute("DELETE FROM {} WHERE App = '{}'".format(uname, app))
    data.commit()
```

#This function is used to delete a user and an application from the database respectively

```
def check(uname, upass):
    with open("data.txt", "r") as fl:
        lst = fl.readlines()
        for i in lst:
            lstel = i.split(',')
            lstel[1] = lstel[1].rstrip().lstrip()
            if lstel[0] == uname:
                if lstel[1] == upass:
                    return True
                else:
    return False
        else:
            return False
```

#This function checks if the username and password entered match the ones given at the time of registration.

```
def ucheck(uname):
    with open("data.txt", "r") as fl:
        lst = fl.readlines()
        for i in lst:
            lstel = i.split(',')
            if uname == lstel[0]:
                print("Username already taken")
                return False
            else:
                pass
        else:
            return True
```

#This function checks if another user exists with the same name already.

Program Loop

```
while True:
    print("\n \n \n \nWelcome to MyPass - your own password manager.")
    try:
        fl = open("data1.txt", "r")
        lst = fl.read().split(',')
        init(lst[0], lst[1])
        fl.close()
    except:
        print("To proceed further, MyPass requires user's MySQL Database
user and password")
        fl = open("data1.txt", "w")
        datauser = input("Enter MySQL database username ")
        datapass = input("Enter MySQL database password ")
        init(datauser, datapass)
        fl.write(datauser+","+datapass)
        fl.close()
    print("What do you want to do?")
    s1 = input("Do you already have a registered username? (y/n) ")
    if s1 == "y":
        uname = input("Enter Username ")
        upass = input("Enter Password ")
        if check(uname, upass):
            print("Select action \n")
        else:
            print(" Invalid username or password")
            continue
 elif s1 == "n":
        while True:
            print("\n \n \nLet\'s create a username")
            uname = input("Enter username ")
            if ucheck(uname) == True:
                upass = input("Enter password ")
                if passstr(upass):
                    pass
                else:
                    print("Enter a password with atleast 8 characters, atleast one lowercase and uppercase letter, atleast one number and atleast one special character([ _ @ $ * ])")
                    continue
                create(uname, upass)
                break
            else:
                continue
    else:
        print("Invalid choice")
        continue
```

#The program starts with a greeting and asking if the user is registered or not. If they are, they'll be taken to the option of actions else will get an option to register themselves.

```
 crtable(uname)
    while True:
        print("\n \nWelcome,", uname)
        print("\n \nWhat do you want to do?")
        print(
            """\n1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7 """)
        choice = int(input())
```

#Once logged in, the program gives the user various functions they can do, like add an application and it's password, modify a password, delete it etc.

```
if choice == 1:
        app = input("Enter app name ")
        appuser = input("Enter app username ")
        pswrd = input("Enter your password on the app ")
        try:
            fenc(pswrd, app, uname, True, appuser)
        except:
            print("App already exists or App name invalid")
elif choice == 2:
        try:
            cursor.execute("SELECT App FROM {}".format(uname))
            lst = cursor.fetchall()[0]
        except:
            print("No app found for registered username")
            continue
        print("Available apps")
        for i in lst:
            print(i)
        app = input("Enter app name ")
        try:
            fdec(app, uname)
        except:
```

```
    print("Invalid app name")
#The basic two functions are directly responded to by the program.


elif choice == 3:
    try:
        cursor.execute("SELECT App FROM {}".format(uname))
        lst = cursor.fetchall()[0]
    except:
        print("No app found for registered username")
        continue
    print("Available apps")
    for i in lst:
        print(i)
    app = input("Enter App to be considered ")
    cursor.execute("SELECT AppUser FROM {} WHERE App =
'{}'".format(uname, app))
    appuser = cursor.fetchone()[0]
    cursor.execute('SELECT keyg FROM {} where App =
"{}"'.format(uname, app))
    key = cursor.fetchone()[0]
    dkey = key.encode()
    f = Fernet(dkey)
```

```
        cursor.execute('SELECT Password FROM {} WHERE App =
"{}"'.format(uname, app))
        pss = cursor.fetchone()[0].encode()
        pswrd = f.decrypt(pss)
        choice1 = int(input("Change 1. App  2. Password  3. App Username
4. App, Username and Password  (1/2/3/4)"))
        if choice1 == 1:
            pswrdnew = pswrd
            appusernew = appuser
            appnew = input("Enter new App name ")
            mod(uname, app, appnew, pswrd, pswrdnew, appuser,
appusernew)
        elif choice1 == 2:
            appnew = app
            appusernew = appuser
            pswrdnew = input("Enter password ")
            mod(uname, app, appnew, pswrd, pswrdnew, appuser,
appusernew)
        elif choice1 == 3:
            appnew = app
            pswrdnew = pswrd
            appusernew = input("Enter new App username")
            mod(uname, app, appnew, pswrd, pswrdnew, appuser,
appusernew)
        elif choice1 == 4:
            appnew = input("Enter new App name ")
            appusernew = input("Enter new App username")
            pswrdnew = input("Enter password ")
            mod(uname, app, appnew, pswrd, pswrdnew, appuser,
appusernew)
        else:
            print("Invalid choice")
```

```
#If the user wants to modify any particular, they can choose the
item they want to change will be decrypted and re-encrypted in
case of passwords.
```

```
 elif choice == 5:
        confirm = input("Are you sure you want to delete {}? (y/n)
".format(uname))
        if confirm == 'y':
            fl = open("data.txt", "r+")
            lst = fl.readlines()
            for i in lst:
                if i.split(",")[0] == uname:
                    lst.pop(lst.index(i))
                    break
                else:
                    pass
            fl.close()
            g = open("data.txt", "w")
```

```
                g.writelines(lst)

                g.close()

                deluser(uname)

                print("Deleted")

                break

            else:

                pass

elif choice == 4:

            delete = input("Enter app name to delete ")

            confirm = input("Are you sure you want to delete {}? (y/n)
".format(delete))

            if confirm == 'y':

                delapp(uname, delete)

                print("Deleted")

            else:

                pass
```

#The program deletes the information about the application or the entire user data if the user wants to.

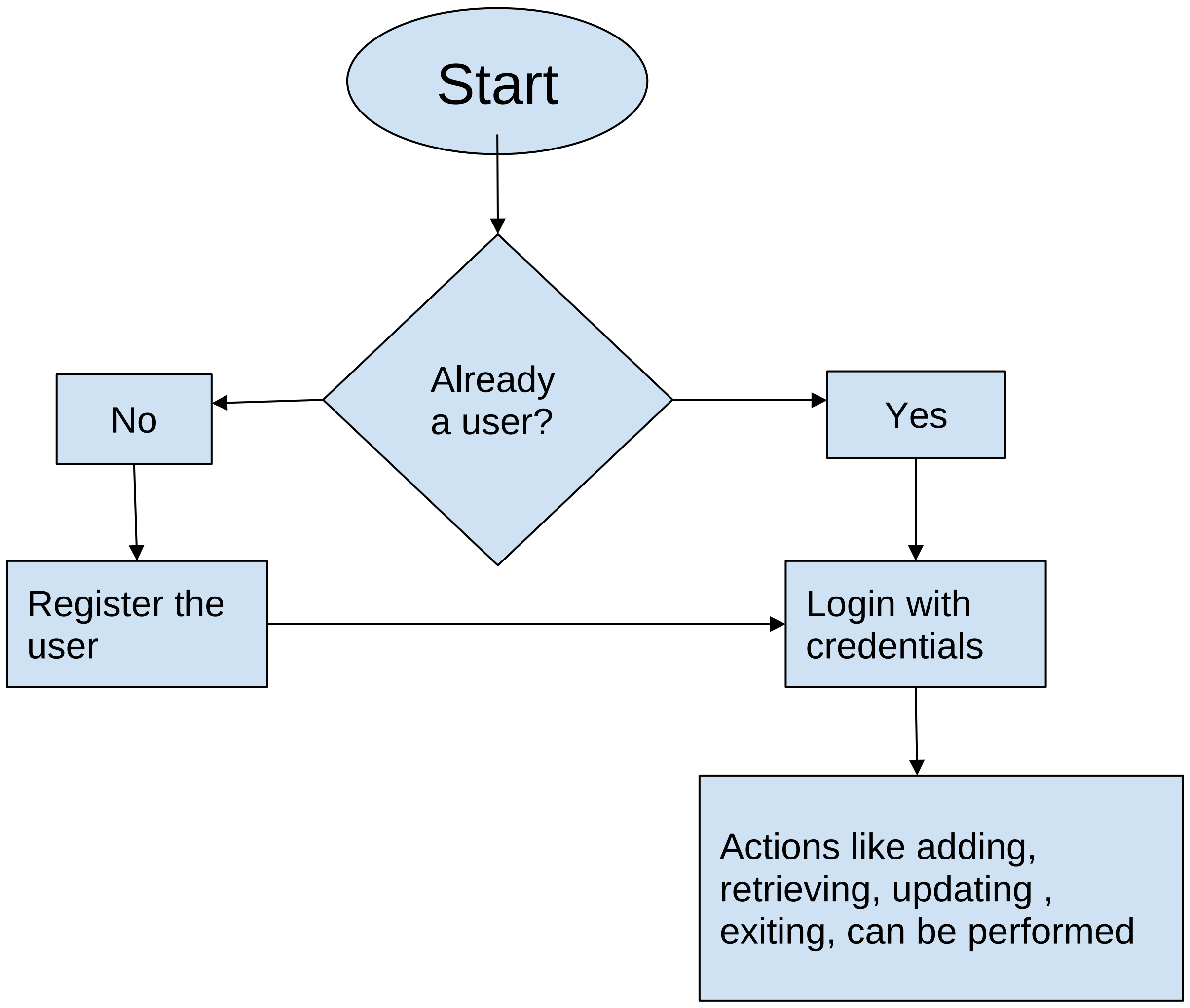
```
          pass

        elif choice == 6 or choice == 7:

            break

        else:
```

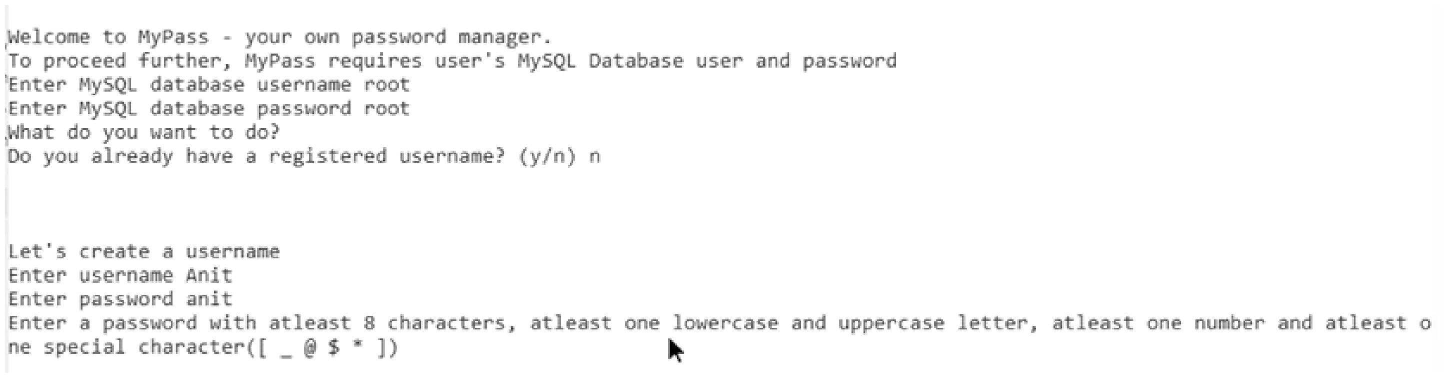
```
        print("Invalid choice")
  if choice == 7:
      break
```

#The program logs out from the user and terminates after the user chooses to do so.

Output

Flow of the program


Start
Already a user?
No
Yes
Register the user
Login with credentials
Actions like adding, retrieving, updating , exiting, can be performed



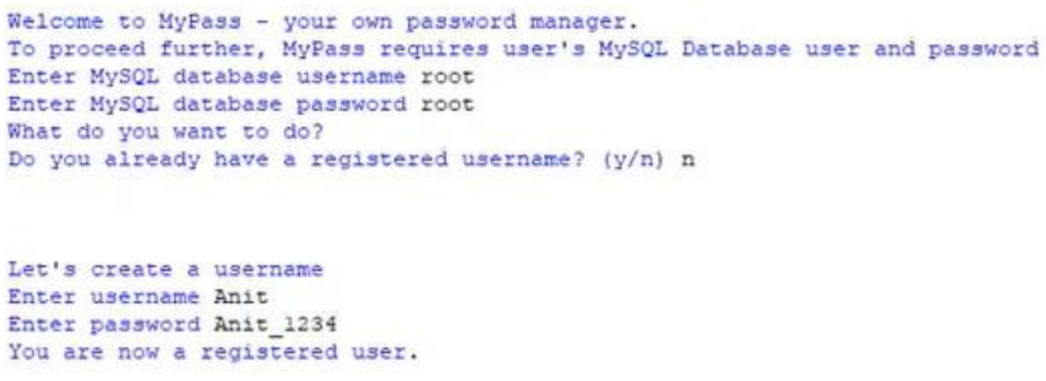
```
Welcome to MyPass - your own password manager.
To proceed further, MyPass requires user's MySQL Database user and password
Enter MySQL database username root
Enter MySQL database password root
What do you want to do?
Do you already have a registered username? (y/n) n


Let's create a username
Enter username Anit
Enter password anit
Enter a password with atleast 8 characters, atleast one lowercase and uppercase letter, atleast one number and atleast o
ne special character([ _ @ $ * ])
```

The first time the program is executed, it asks the user to login to the MySQL database with their credentials where their data will be saved after encryption.

The program then asks if the user is already registered so that it can access the old data of that user, else asks the user to register themselves into the program.

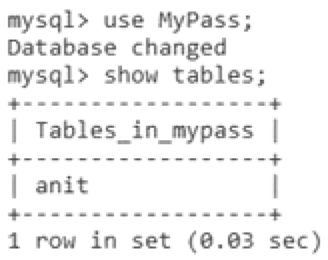
The password entered should be a strong password with at least 8 characters, one lowercase character, one uppercase character, one number and at least one special character.

If the conditions are not satisfied the program shows an error.



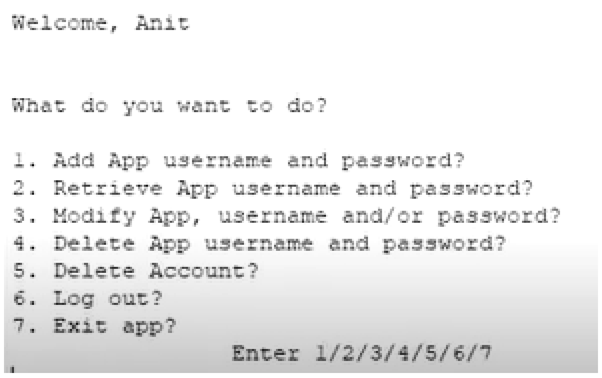
```
Welcome to MyPass - your own password manager.
To proceed further, MyPass requires user's MySQL Database user and password
Enter MySQL database username root
Enter MySQL database password root
What do you want to do?
Do you already have a registered username? (y/n) n


Let's create a username
Enter username Anit
Enter password Anit_1234
You are now a registered user.
```

When a valid password is entered, the user gets registered and can then add, access, modify, delete the data entered by the user after encryption and decryption.



```
mysql> use MyPass;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_mypass |
+-------------------+
| anit             |
+-------------------+
1 row in set (0.03 sec)
```

When we open the MySQL database, a new table can be seen of the new user under which, when opened, will be containing the encrypted passwords with their username and a key to decrypt it later when accessed.



```
Welcome, Anit


What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7
```

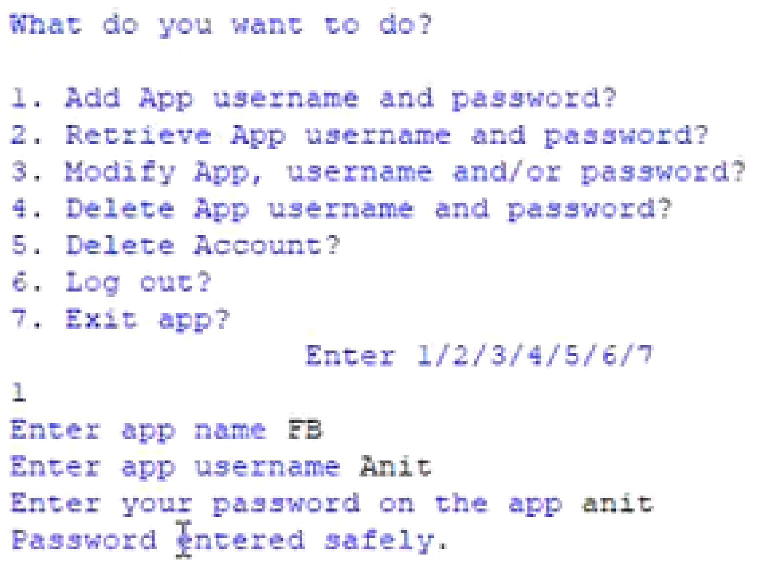
Once the user logs in, the user gets many options to take up. The user can add different sites and passwords to them.

They can retrieve those details anytime they want. They can also change the details like passwords or the username too. They can also delete different content in it.

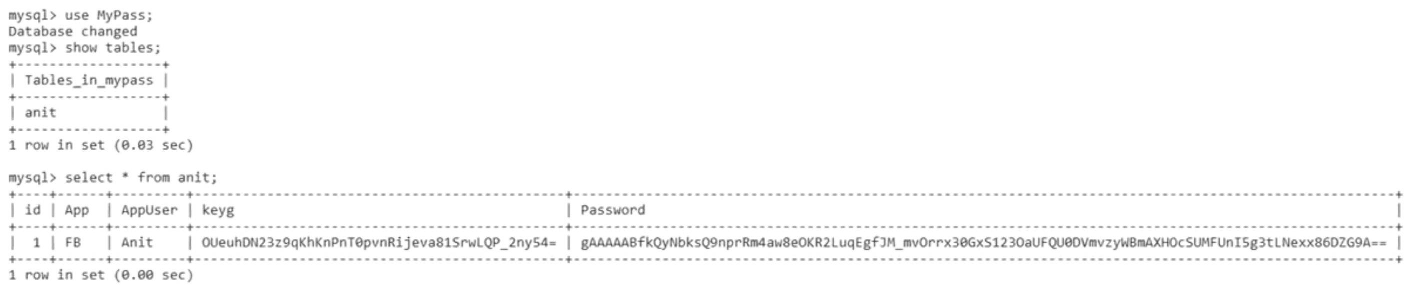
There is also an option to delete the account itself which will delete all the passwords of different apps too.

The log out option logs the user out and other users can log in and save their own passwords separately.

The user can exit the program by choosing the last option.

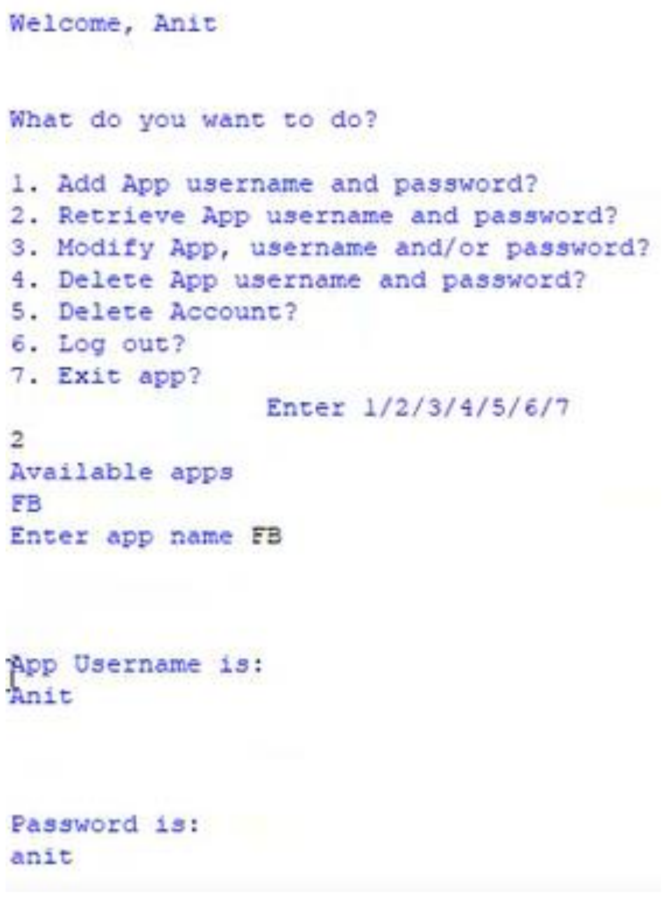


```
What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                    Enter 1/2/3/4/5/6/7
1
Enter app name FB
Enter app username Anit
Enter your password on the app anit
Password entered safely.
```



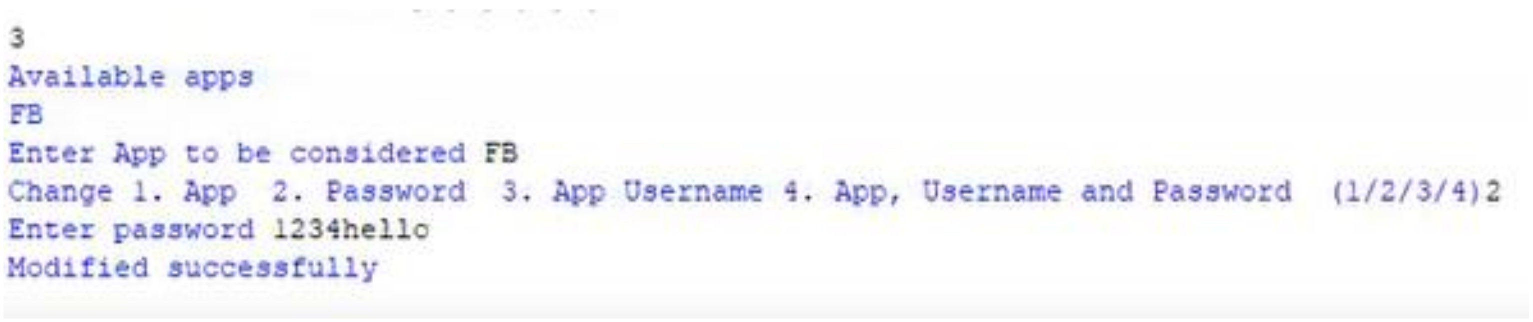
```
mysql> use MyPass;
Database changed
mysql> show tables;
+-----------------+
| Tables_in_mypass |
+-----------------+
| anit            |
+-----------------+
1 row in set (0.03 sec)

mysql> select * from anit;
+----+------+---------+----------------------------------------------+------------------------------------------------------------------------------------------------------+
| id | App  | AppUser | keyg                                         | Password                                                                                             |
+----+------+---------+----------------------------------------------+------------------------------------------------------------------------------------------------------+
|  1 | FB   | Anit    | OUeuhDN23z9qKhKnPnT0pvnRijeva81SrwLQP_2ny54= | gAAAAABfkQyNbksQ9nprRm4aw8eOKR2LuqEgfJM_mvOrrx30GxS123OaUFQU0DVmvzyWBmAXHOcSUMFUnI5g3tLNexx86DZG9A== |
+----+------+---------+----------------------------------------------+------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

When the app name and password is entered by the user, the program encrypts the password and saves it into the MySQL database with the key to decrypt it later.



```
Welcome, Anit


What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7
2
Available apps
FB
Enter app name FB



App Username is:
Anit



Password is:
anit
```

When the user wants to access their username and passwords, the program lists down all the apps that have been added in the database. The user then chooses one of the apps and the program decrypts the password and gives the username and password to the user.



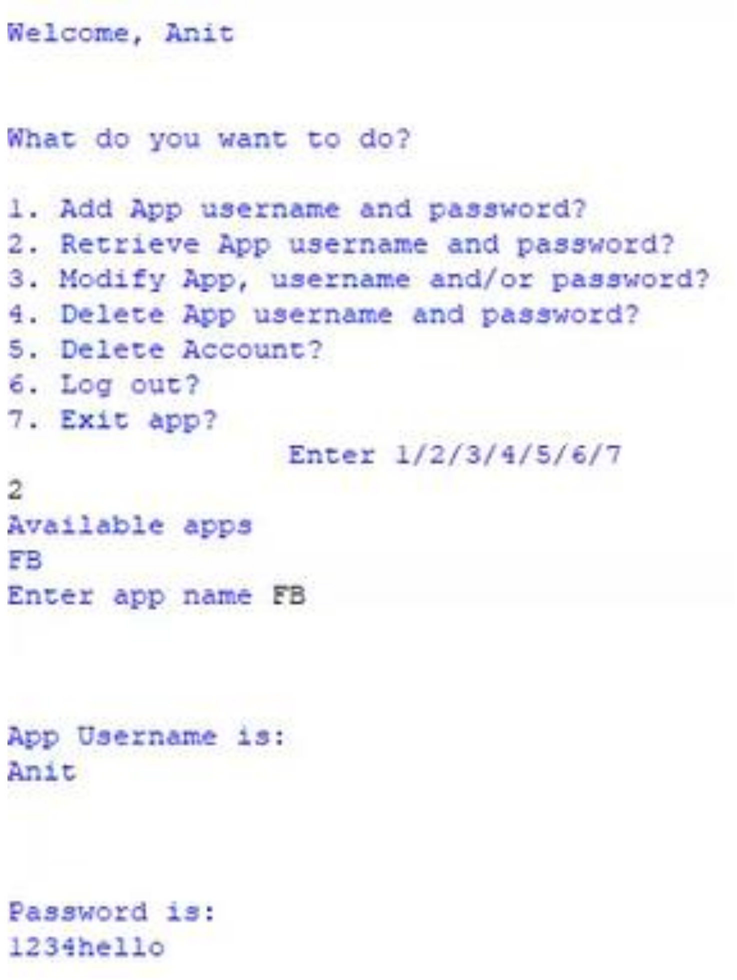
```
3
Available apps
FB
Enter App to be considered FB
Change 1. App  2. Password  3. App Username 4. App, Username and Password  (1/2/3/4)2
Enter password 1234hello
Modified successfully
```

When the user chooses the option to modify the details, the program lists down all the names of the different apps of which the details can be changed.

Once the user chooses the app of which they wish to change the details of, the program gives the options of the things they want to change about that app.

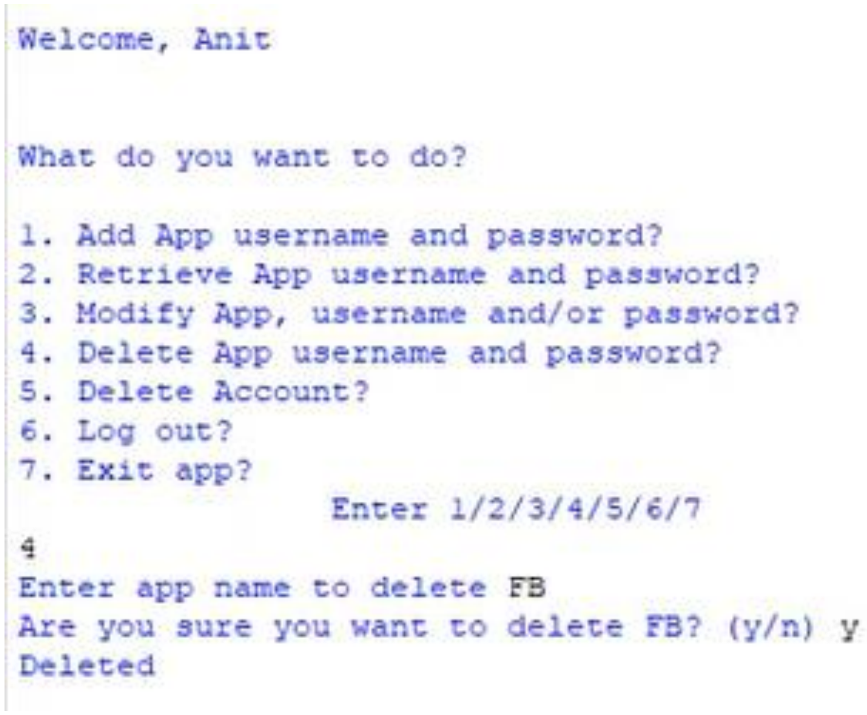
The user can choose from changing the app name, the password, the username of the app, or everything at once.

Once the details are changed, the program passes the statement “Modified successfully”



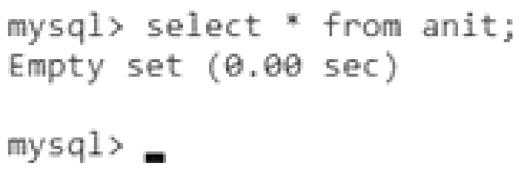
```
Welcome, Anit


What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7
2
Available apps
FB
Enter app name FB



App Username is:
Anit




Password is:
1234hello
```

When the password is accessed after modifying the details, the new updated password is displayed. It is also saved in an encrypted manner with a key in the same database and same table.



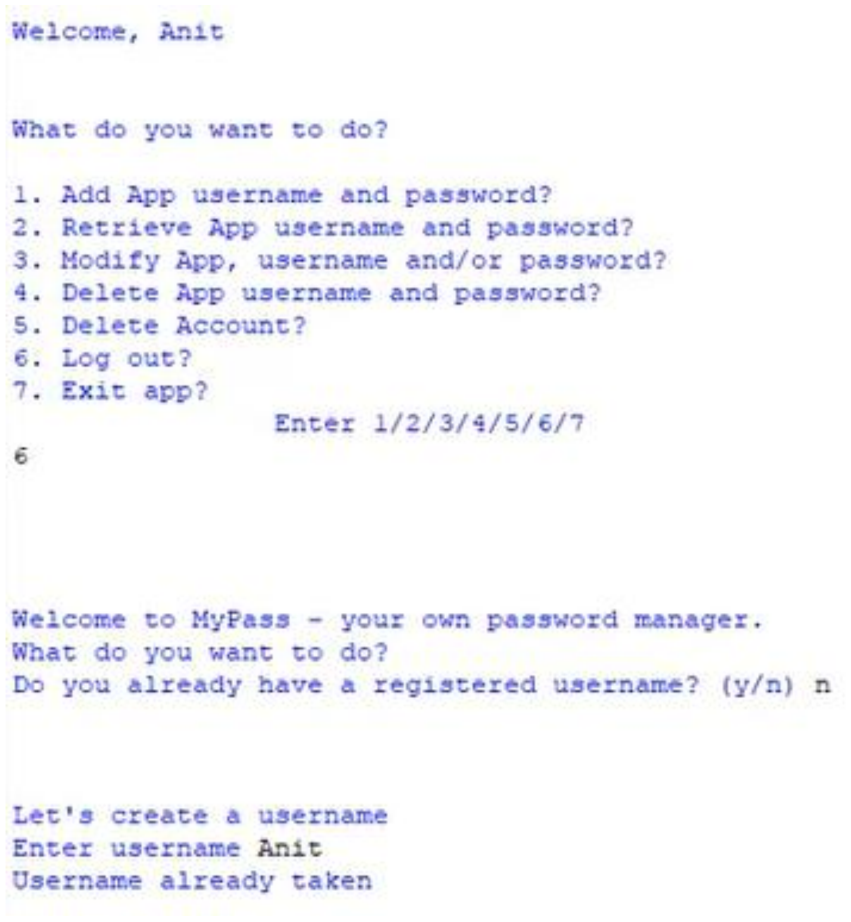
```
Welcome, Anit


What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7
4
Enter app name to delete FB
Are you sure you want to delete FB? (y/n) y
Deleted
```

The user can also delete anything they want to.



```
mysql> select * from anit;
Empty set (0.00 sec)

mysql> _
```

Once the user deletes anything through the program, the database instantly updates and that part deletes from there too.



```
Welcome, Anit


What do you want to do?

1. Add App username and password?
2. Retrieve App username and password?
3. Modify App, username and/or password?
4. Delete App username and password?
5. Delete Account?
6. Log out?
7. Exit app?
                Enter 1/2/3/4/5/6/7
6




Welcome to MyPass - your own password manager.
What do you want to do?
Do you already have a registered username? (y/n) n



Let's create a username
Enter username Anit
Username already taken
```

The log out option logs out the user and the program starts from the top.

The new user cannot be of the same name as others already taken.

The exit option kills the program there and also logs out the user. The database cannot be accessed in any way other than this program or the login credentials of the database directly.

Bibliography

1. COMPUTER SCIENCE WITH PYTHON Class XII by Sumita Arora
2. w3schools.com
3. devqa.io
4. stackoverflow.com

MyPass