

**V.K.K MENON COLLEGE Of COMMERCE  
AND SCIENCE**

**Bhandup (East), Mumbai-42**

**Department Of Computer Science**

**A  
PROJECT REPORT ON  
“PASSWORD MANAGER”**

**BY**

***Mr. Roghan Bharadwaj***

**ACADEMIC YEAR:2021-2022**

**UNDER THE GUIDANCE OF**

**Dr. Saloni Bhushan**

**Mr. Rajesh Yadav**

**SUBMITTED FOR PARTIAL FULFILLMENT OF B.Sc.  
DEGREE COURSE IN COMPUTER SCIENCE OF  
UNIVERSITY OF MUMBAI**

**CERTIFICATE**

**V.K KRISHNA MENON COLLEGE OF COMMERCE AND ECONOMICS AND  
SHARAD SHANKAR DIGHE COLLEGE OF SCIENCE  
BHANDUP EAST, MUMBAI - 400042**

This is to certify that Mr/~~Miss~~ ROSHAN BHARADWAJ

Seat No 04 has successfully completed the PROJECT titled PASSWORD MANAGER

for complete fulfilment of Bachelor of Computer Science of University of Mumbai in

academic year 2021 - 2022 under the guidance of Dr. Saloni Bhushan

Mr. Rajesh Yadav

DATE:

HEAD OF THE DEPARTMENT

PROJECT GUIDE

EXAMINER

Dr. Saloni Bhushan

Mr. Rajesh Yadav

**INDEX**

Sr No.	Topics	Page No.
01	<b>Acknowledgement</b>	
02	<b>Overview</b>	
a	Abstract	
b	Undertaking	
c	Objectives	
d	Introduction	
e	Requirement Specification	
f	Feasibility Study	
g	Gantt chart	
03	<b>System Design</b>	
a	ER Diagram	
b	Class Diagram	
c	Use Case Diagram	
d	Sequence Diagram	

04	<b>Code Implementation</b>	
a	code	
b	Screen Shots	
05	<b>Future Scope</b>	
06	<b>Limitation</b>	
07	<b>Reference</b>	

### **ACKNOWLEDGEMENT**

I am grateful to Mumbai University for introducing such practical projects that let students learn form more than just books.

It gives me immense pleasure as I present to you my project on "Password Manager". This acknowledgement is a small effort to express my gratitude to all those who have shown me the path to bring out the various colors of this project with their vast treasure of experience and knowledge.

I would like to express my sincere thanks to Prof. **Mr. Rajesh Yadav**. who helped me throughout the project by providing me moral support, A conducive work environment and the much-needed inspiration to conclude this project on time

I also take this opportunity to thank our Head of the Department **Dr.Saloni Bhushan** and also all the professors of the Department of Computer Science of V.K Krishna Menon College for giving me an opportunity to study in the institute and the most needed guidance throughout the duration of the course.

Prof. **Mr. Rajesh Yadav** and **Dr. Saloni Bhushan** have provided me with the guidance and necessary support during each phase of the project. They were the Source of continuous encouragement as each milestone was crossed.

My parents were my first teachers and they have provided me with such a great exposure that has helped me bloom. My family and friends will always be loved for sticking by me through thick and thin. My lovely classmates who have chipped in with their aid and suggestions, THANK YOU!

**UNDERTAKING**

This is to declare that the project titled “**Password Manager**” is an original work done by the undersigned, in partial fulfilment of the requirement for the degree “Bachelor of Science in Computer Science” at CS Department, V. K. Krishna Menon College. All the analysis design and System developed have been accomplished by the undersigned. Moreover, this project has not been submitted to any other College or University.

**Mr. Roshan Bhardwaj**

## **OBJECTIVE:**

---

“Password Manager” is Python application which stores account details with website link, username and password all in one place. Password is stored in encrypted form and can only be decrypted by key. So that instead of using same password again and again we can use strong passwords for different websites and we don’t have to remember those passwords because we can get all the passwords from “password manager” with one key.

## **ABSTRACT**

The project “password manager” is password manager application which saves your account information with encryption it saves passwords in encrypted form

Which cant be accessed without key.

## **INTRODUCTION**

---

“Password Manager” is Python Tkinter application which provides safe place your all online account details in one place which can be accessed by just one secret key

And we can add new accounts or update existing account details. It also provide features like redirect to the website of current selected account or copy the password of selected account or we can also delete the selected account.

With search feature we can filter our account by website name or username and it will show all the account related to websites or username. With clear button we can clear all the filters

“Password Manager” also provides feature to reset your secret key with the help of security question in case user forget secret key.

## **REQUIREMENTS**

---

### **Software Requirements**

- Operating System                      Windows,Linux
- Programming Language              Python
- external libraries                      clipboard
- Hardware Requirements
  - Desktop PC, Laptop.
  - Operating System – Windows and Linux.
  - Hard Disk 1GB
  - Keyboard and Mouse



## **FEASIBILITY STUDY**

---

### Technical Feasibility

No Specialized equipment's or hardware are required to run this password manager

It produces minimal strain upon the system making it feasible to use regardless of the system specifications as long as the system can run python script.

Build is possible cross platform with any IDE or text editor making it seamless to use across python supported platform as well as makes its build process easy. It requires Python which is supported on all operating systems.

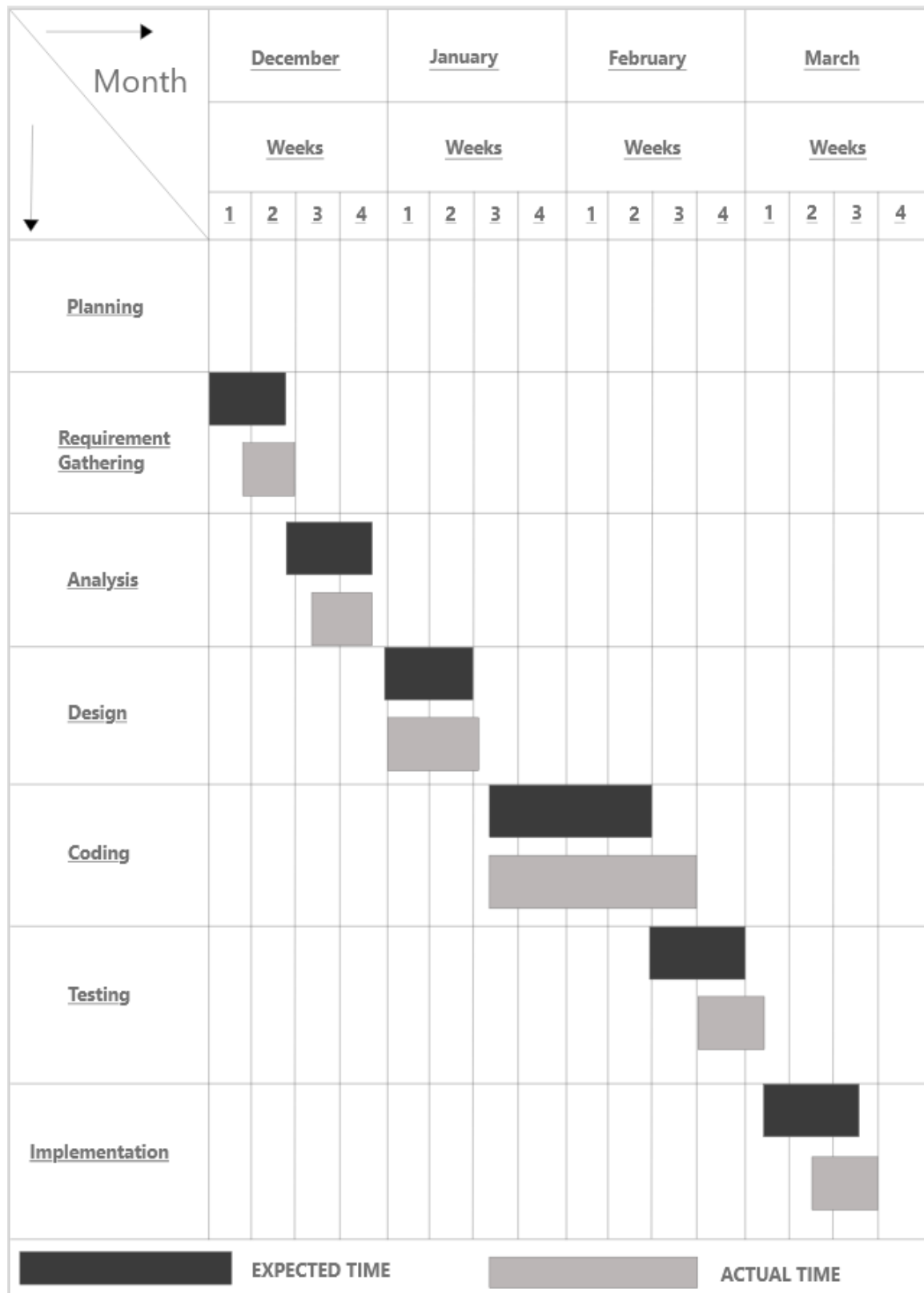
### Financial Feasibility

There is no initial investment required of any form. As long as you have a simple and sober system you could run and also build the software

There is no need for any unique equipment's or hardware, so it is not financially constrained by hardware.

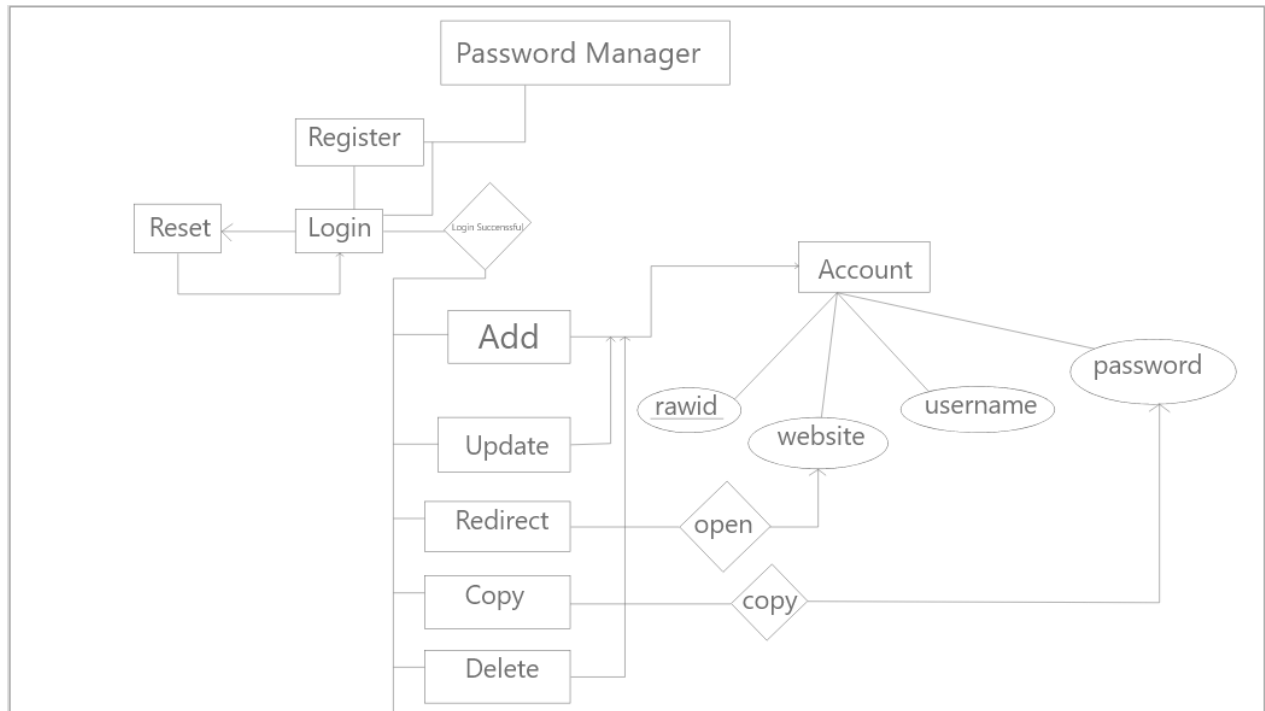
No initial resources or any licenses required to run.

### Gantt Chart



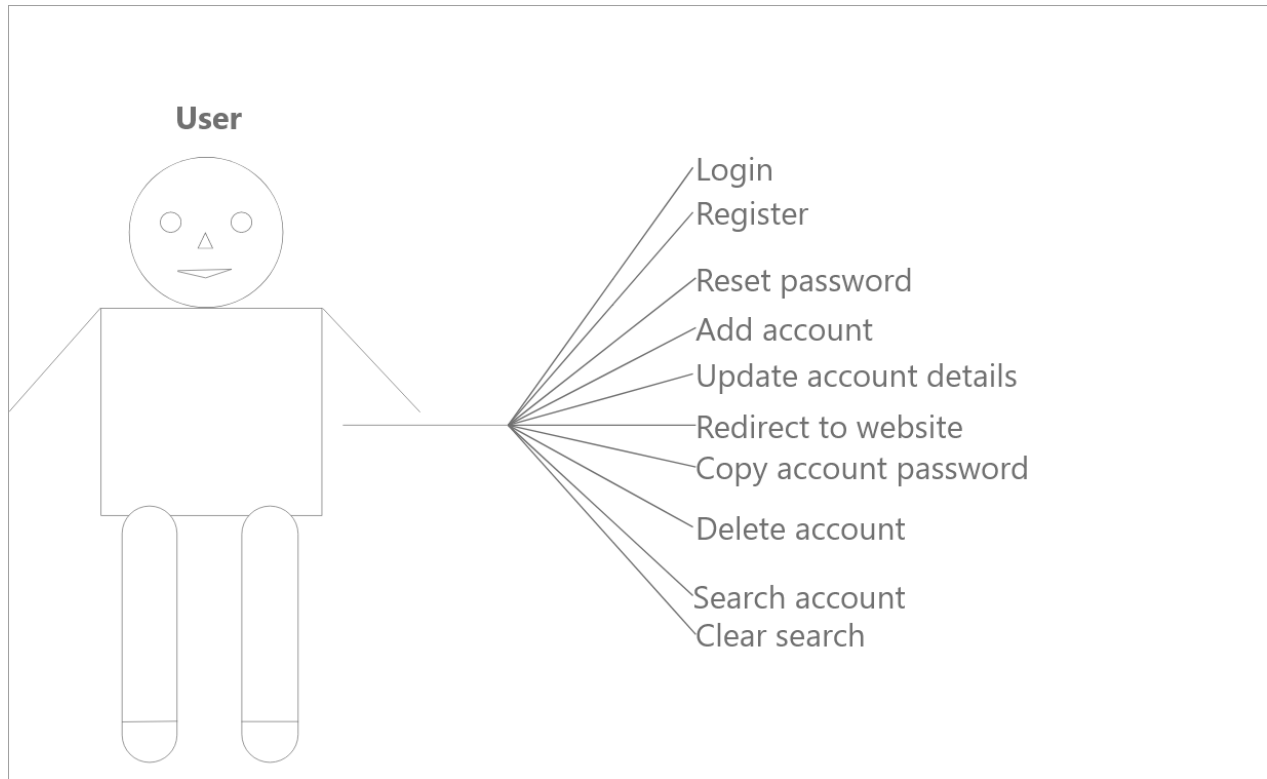
## SYSTEM DESIGN DETAILS

### ER DIAGRAM:-

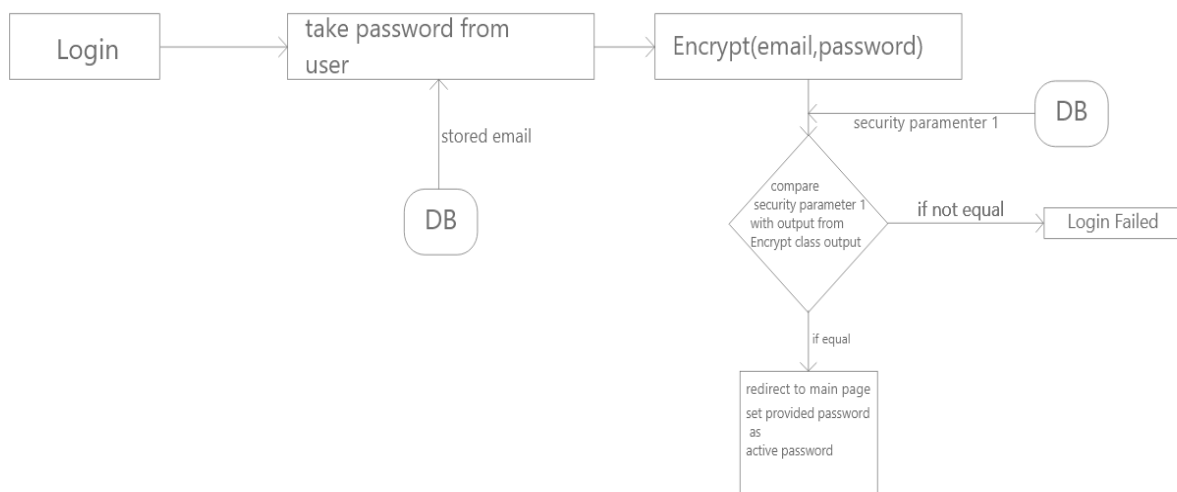
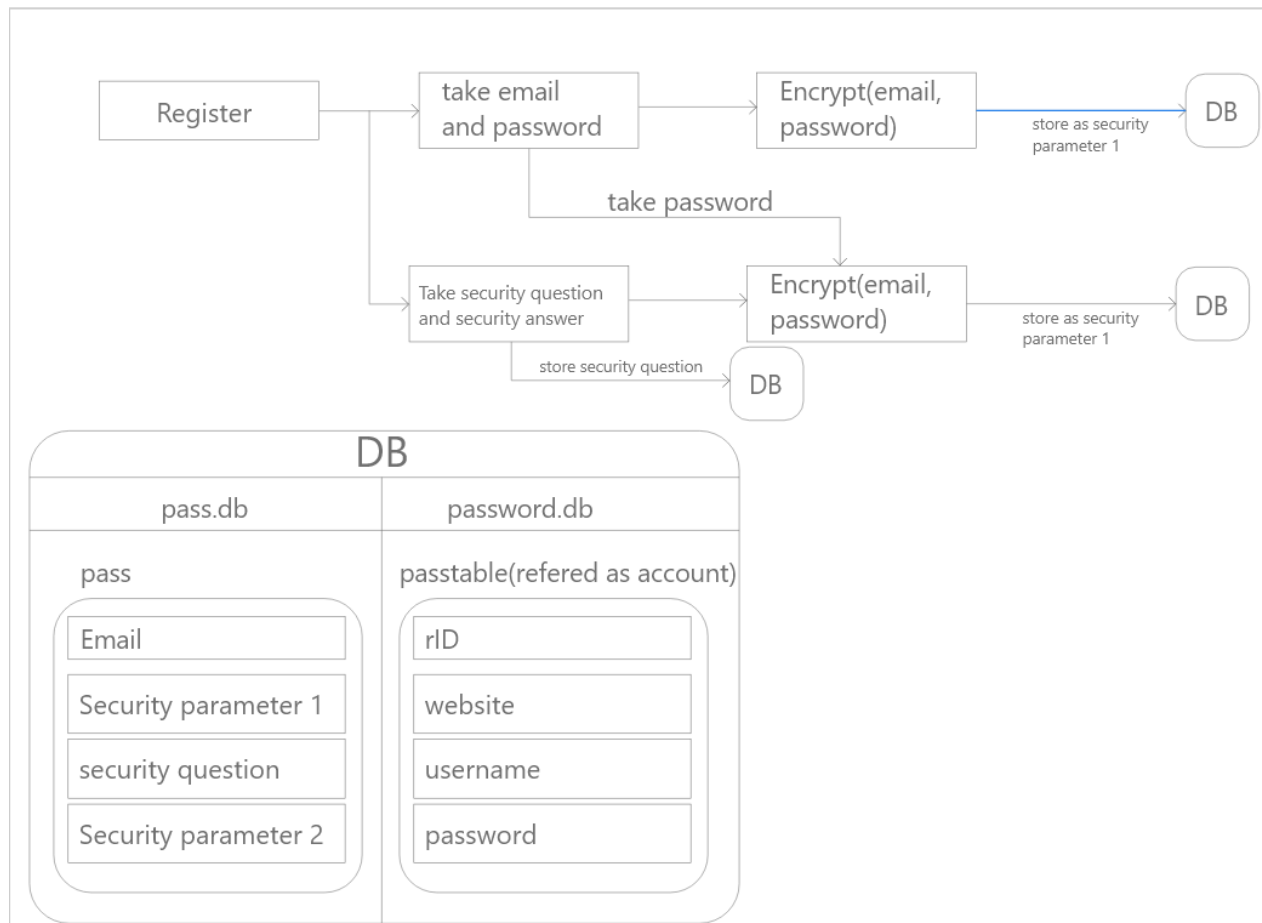


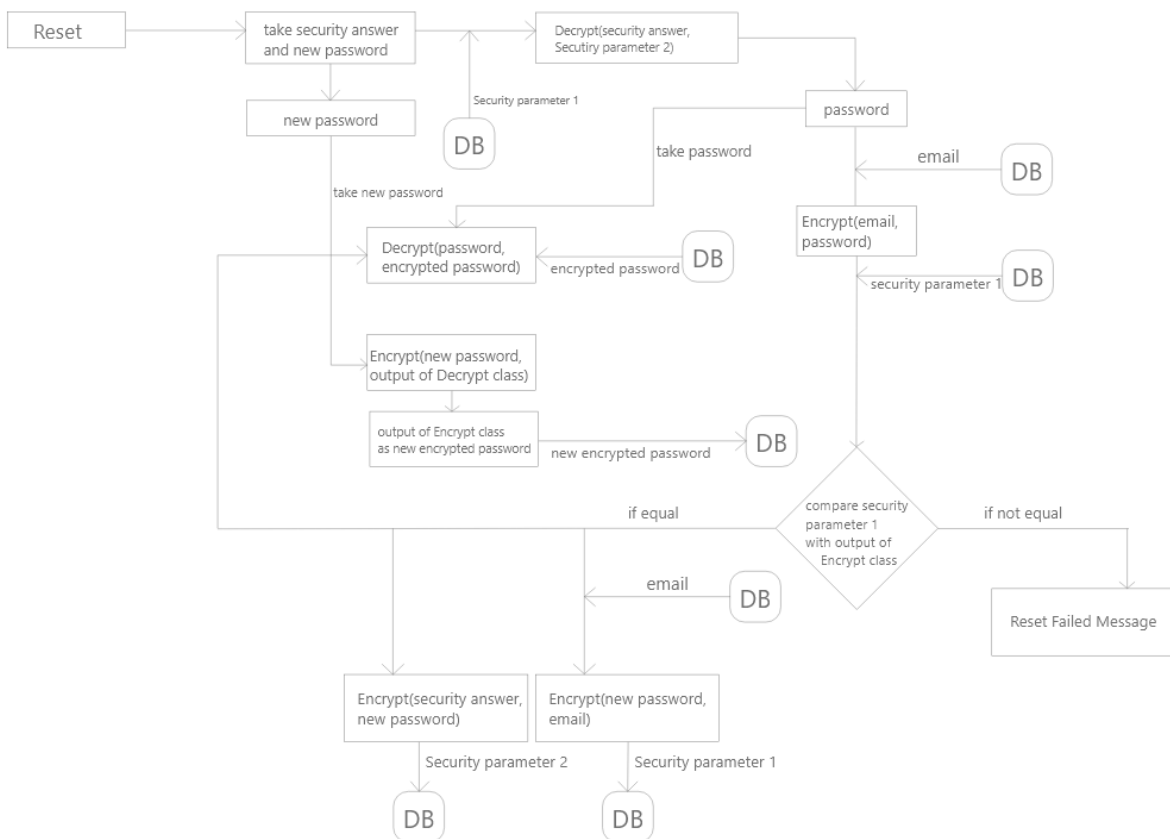
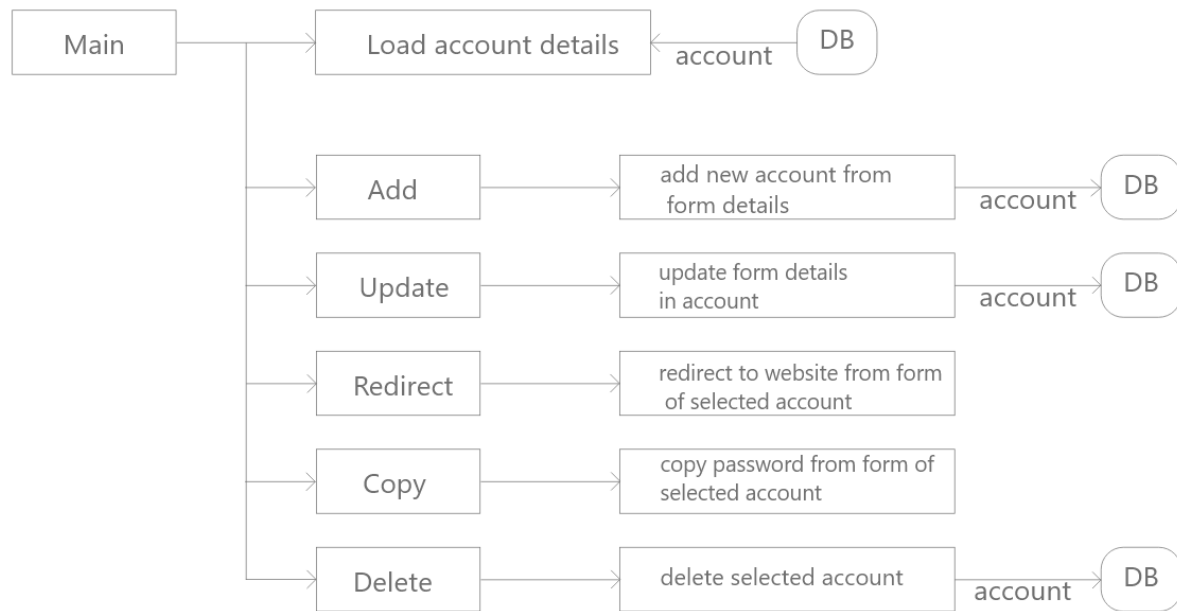
### CLASS DIAGRAM:-



**USE CASE DIAGRAM:-**

## Sequence diagram





## CODE IMPLEMENTATIONS

```
from tkinter import *
from tkinter import messagebox
from tkinter.ttk import Treeview
from PIL import ImageTk, Image
import sqlite3

import re

import webbrowser

import clipboard

active_pass=""

class Login:

    def __init__(self) -> None:
        def resetPass():
            self.login.destroy()
            Reset()

        self.login=Tk()
        self.login.title("Login")
        self.login["bg"]="#ff972e"
        self.login.geometry("600x400")

        self.contents=Frame(self.login)
        self.contents["bg"]="white"
        self.contents.pack(fill="both", expand="yes", padx=10, pady=38)

        self.text=Label(self.contents, text="Welcome to Password
Manager", font=("Arial", 20))
        self.text.place(x=100, y=50)

        self.password=Entry(self.contents, show="*", font=("Arial", 18))
        self.password.insert(0, "password")
        self.password.place(x=300, y=200, anchor=CENTER)

        self.loginBtn=Button(self.contents, text="Login", command=self.checkpass)
        self.loginBtn.place(x=300, y=250, height=30, width=100, anchor=CENTER)
```

```
self.resetBtn=Button(self.contents,text="Reset
Password",command=resetPass)
self.resetBtn.place(x=300,y=300,height=30,width=100,anchor=CENTER)

self.login.mainloop()

def checkpass(self):
    global active_pass
    conn=sqlite3.connect("pass.db")
    cur=conn.cursor()
    storedData=cur.execute("select email,password from pass")
    storedEmail=""
    storedPassword=""
    for row in storedData:
        storedEmail=row[0]
        storedPassword=row[1]
    if storedPassword ==
Encryption(self.password.get(),storedEmail).getEncryption():
        active_pass=self.password.get()
        self.login.destroy()
        Main()
    else:
        self.text.config(text="Login Failed Try Again")

class Main:

    def __init__(self) -> None:

        def addNew():
            if self.passwordText.get()!="":
                if
database.newRecord((self.websiteText.get(),self.usernameText.get(),Encryption(active_pass,self.passwordText.get()).getEncryption())):
                    messagebox.askokcancel("Successfull","New Record Created
Successfully")
                clear()
            else:
                messagebox.askokcancel("Failed","Password can't be empty")

        def update(data):
            trv.delete(*trv.get_children())
            for row in data:
                trv.insert("",END,values=row)

        def search():
```



```
sT=searchText.get()
conn=sqlite3.connect("password.db")
cur=conn.cursor()
data=cur.execute("SELECT * FROM passtable WHERE website LIKE
'%" +sT+"%' OR username LIKE '%" +sT+"%' ")
update(data)
conn.close()

def clear():
    conn=sqlite3.connect("password.db")
    cur=conn.cursor()
    data=cur.execute("SELECT * FROM passtable")
    update(data)
    conn.close()

def getDetails(event):
    rowid=trv.identify_row(event.y)
    item=trv.item(trv.focus())
    self.accountNo.set(str(item['values'][0]))
    self.websiteText.set(str(item['values'][1]))
    self.usernameText.set(str(item['values'][2]))
    password=Description(str(active_pass),str(item['values'][3])).getDecr
ption()
    self.passwordText.set(password)

def updateDetails():
    if self.passwordText.get()!="":
        if
database.update((self.websiteText.get(),self.usernameText.get(),Encryption(active
_pass,self.passwordText.get()).getEncryption(),self.accountNo.get())):
            messagebox.askokcancel("Update Successfull","The account
details has been updated")
            clear()
        else:
            messagebox.askokcancel("Failed to update account
details","Please check all the details and try again")
        else:
            messagebox.askokcancel("Failed","Password can't be empty")

def redirect():
    webbrowser.open(self.websiteText.get(), new=2)

def copyPass():
    clipboard.copy(self.passwordText.get())
```

```
        messagebox.askokcancel("Copy Successfull","Password copy to clipboard
successfull")

    def deleteAccount():
        if self.accountNo.get()!="":
            and messagebox.askyesno("Confirm
Delete?","Are you sure you want to delete this account?"):
                print(self.accountNo.get())
                database.delete(self.accountNo.get())
                clear()
            else:
                messagebox.askokcancel("Account Id Not Available","Double click
on record then delete")

    self.main=Tk()
    self.main.title("Accounts")
    self.main.geometry("800x600")
    self.main["bg"]="#ff972e"
    searchText=StringVar()

    self.accountNo=StringVar()
    self.websiteText=StringVar()
    self.usernameText=StringVar()
    self.passwordText=StringVar()

    #icons
    Iadd=ImageTk.PhotoImage(Image.open("icons/add.png"))
    Isave=ImageTk.PhotoImage(Image.open("icons/save.png"))
    Iredirect=ImageTk.PhotoImage(Image.open("icons/redirect.png"))
    Icopy=ImageTk.PhotoImage(Image.open("icons/copy.png"))
    Idelete=ImageTk.PhotoImage(Image.open("icons/delete.png"))

    wrapper1=LabelFrame(self.main,text="Account List")
    wrapper2=LabelFrame(self.main,text="search")
    wrapper3=LabelFrame(self.main,text="Account Data")

    wrapper1.pack(fill="both",expand="yes",padx=20,pady=10)
    wrapper2.pack(fill="both",expand="yes",padx=20,pady=10)
    wrapper3.pack(fill="both",expand="yes",padx=20,pady=10)

    trv=Treeview(wrapper1,columns=(1,2,3,4),show="headings",height="6")
    trv.pack()

    trv.heading(1,text="No")
    trv.heading(2,text="Website")
```

```
trv.heading(3,text="Username")
trv.heading(4,text="Password")

trv.bind('<Double 1>',getDetails)

conn=sqlite3.connect("password.db")
cur=conn.cursor()
data=cur.execute("SELECT * FROM passtable")
update(data)
conn.close()

#search Section
lbl=Label(wrapper2,text="search")
lbl.pack(side=LEFT,padx=10)
searchEntry=Entry(wrapper2,textvariable=searchText)
searchEntry.pack(side=LEFT,padx=6)
searchBtn=Button(wrapper2,text="search",command=search)
searchBtn.pack(side=LEFT,padx=6)
clearBtn=Button(wrapper2,text="Clear",command=clear)
clearBtn.pack(side=LEFT,padx=6)

#Edit section
Lwebsite=Label(wrapper3,text="Website")
Lwebsite.grid(row=0,column=0,padx=5,pady=3)
Ewebsite=Entry(wrapper3,textvariable=self.websiteText)
Ewebsite.grid(row=0,column=1,padx=5,pady=3)

Lwebsite=Label(wrapper3,text="Username")
Lwebsite.grid(row=1,column=0,padx=5,pady=3)
Ewebsite=Entry(wrapper3,textvariable=self.usernameText)
Ewebsite.grid(row=1,column=1,padx=5,pady=3)

Lwebsite=Label(wrapper3,text="Password")
Lwebsite.grid(row=2,column=0,padx=5,pady=3)
Ewebsite=Entry(wrapper3,textvariable=self.passwordText)
Ewebsite.grid(row=2,column=1,padx=5,pady=3)

addBtn=Button(wrapper3,text="+",command=addNew,image=Iadd)
addBtn.grid(row=3,column=0,padx=5,pady=3)

updateBtn=Button(wrapper3,text="^",command=updateDetails,image=Isave)
updateBtn.grid(row=3,column=1,padx=5,pady=3)

redirectBtn=Button(wrapper3,text=">",command=redirect,image=Iredirect)
redirectBtn.grid(row=3,column=2,padx=5,pady=3)
```

```
copyBtn=Button(wrapper3,text="C",command=copyPass,image=Icopy)
copyBtn.grid(row=3,column=3,padx=5,pady=3)

deleteBtn=Button(wrapper3,text="X",command=deleteAccount,image=Idelete)
deleteBtn.grid(row=3,column=4,padx=5,pady=3)

self.main.mainloop()

class database:

    def getall():
        conn=sqlite3.connect("password.db")
        cur=conn.cursor()
        data=cur.execute("SELECT * FROM passtable")
        return data

    def newRecord(values):
        conn=sqlite3.connect("password.db")
        cur=conn.cursor()
        try:
            cur.execute("INSERT INTO passtable(website,username,password)
values(?,?,?)",values)
            conn.commit()
            return True
        except:
            print("failed to insert Record")

    def delete(id):
        conn=sqlite3.connect("password.db")
        cur=conn.cursor()
        deleteRow=cur.execute("DELETE from passtable where rowid=(?)",(str(id),))
        conn.commit()

    def update(values):
        conn=sqlite3.connect("password.db")
        cur=conn.cursor()
        try:
            deleteRow=cur.execute("""UPDATE passtable set
                                website=(?),
                                username=(?),
                                password=(?)
```

```
                where rowid=(?)""",
                values)

        conn.commit()
        return True
    except Exception as e:
        print(e)

class Encryption:
    def __init__(self, key, text) -> None:
        self.key=key.lower()
        self.text=(text+self.key).lower()

    def wordToNum(self):
        self.listOfNum=[]
        for ind,c in enumerate(self.text):
            self.listOfNum.append(ord(c)+ord(self.key[ind%len(self.key)]))

    def decToBin(self):
        self.listOfBin=[]
        for dec in self.listOfNum:
            self.listOfBin.append(f'{dec:b}')
        return self.listOfBin

    def getEncryption(self):
        self.wordToNum()
        binary=self.decToBin()
        return "".join(binary)

class Descryption:

    def __init__(self, key, text) -> None:
        self.key=key.lower()
        self.text=text.lower()
        self.originalString=""

    def stringToBin(self):
        self.listOfBin=[]
        for i in range(0,len(self.text),8):
            self.listOfBin.append(self.text[i:i+8])

    def binaryToNum(self):
        self.listOfDec=[]
        for binary in self.listOfBin:
            self.listOfDec.append(int(binary,2))
```

```
def numToWord(self):
    self.listOfWord=""
    for ind,c in enumerate(self.listOfDec):
        self.listOfWord+=chr(c-ord(self.key[ind%len(self.key)]))

def getDecryption(self):
    self.stringToBin()
    self.binaryToNum()
    self.numToWord()
    return self.listOfWord[:-len(self.key)]

class Register:

    def __init__(self) -> None:
        self.eye=0
        def toggleEye():
            if self.eye==0:
                self.password.config(show="")
                self.eye=1
            else:
                self.password.config(show="*")
                self.eye=0

        self.register=Tk()
        self.register.title("Register")
        self.register["bg"]="#F9A958"
        self.register.geometry("600x400")

        questions=["what is your nickname?","what is your first teacher
name?","what is your bestfriend's name?","what is the name of first movie you
watched?"]

        self.clicked=StringVar()
        self.clicked.set(questions[0])
        self.contents=Frame(self.register)
        self.contents["bg"]="white"
        self.contents["highlightbackground"]="#666262"
        self.contents["highlightthickness"]=7
        self.contents.place(x=100,y=100,height=270,width=380)

        #Icon Refrence
        Ipassword=ImageTk.PhotoImage(Image.open("icons/password.png"))
        Iemail=ImageTk.PhotoImage(Image.open("icons/email.png"))
        Ieye=ImageTk.PhotoImage(Image.open("icons/eye.png"))
```

```
#labels
Lpassword=Label(self.contents,image=Ipassword,bg="white")
Lemail=Label(self.contents,image=Iemail,bg="white")
LwelcomeMsg=Label(self.register,
                    text="Welcome to Password Manager",
                    font=("Arial Rounded MT Bold",20),
                    fg="white",
                    bg="#F9A958")
LsetDetails=Label(self.contents,
                  bg="white",
                  text="Fill the details",
                  font=("Arial Rounded MT Bold",12),
                  fg="#F9A958")

#Input box
self.password=Entry(self.contents,highlightthickness=1,bd=0)
self.password.config(highlightbackground="#F9A958",
highlightcolor="#F9A958",show="*")
self.email=Entry(self.contents,highlightthickness=1,bd=0)
self.email.config(highlightbackground="#F9A958",
highlightcolor="#F9A958")
self.securtyQuestion=OptionMenu(self.contents,self.clicked,*questions)
self.securtyAnswer=Entry(self.contents,highlightthickness=1,bd=0)
self.securtyAnswer.config(highlightbackground="#F9A958",
highlightcolor="#F9A958")

#Button
eyeBtn=Button(self.contents,image=Ieye,bd=0,bg="white",command=toggleEye)
registerBtn=Button(self.contents,
                   text="Register",
                   bg="#4BFA17",
                   fg="white",
                   bd=0,
                   font=("Arial Rounded MT Bold",11),
                   command=self.savePass)

#all Placements
LwelcomeMsg.place(x=98,y=27,height=32,width=428)
LsetDetails.place(x=182-126,y=142-133,height=40,width=212)
Lpassword.place(x=159-126,y=203-133,height=26,width=26)
self.password.place(x=190-126,y=200-133,height=31,width=217)
eyeBtn.place(x=420-126,y=200-133,height=31,width=33)
Lemail.place(x=155-126,y=241-133,height=30,width=30)
```

```
self.email.place(x=190-126,y=238-133,height=31,width=217)
self.securtyQuestion.place(x=190-126,y=275-133,height=31,width=222)
self.securtyAnswer.place(x=190-126,y=310-133,height=31,width=217)
registerBtn.place(x=249-126,y=346-133,height=30,width=79)

self.register.mainloop()

def savePass(self):
    #validation

    #password validation
    if len(self.password.get())<1:
        messagebox.askokcancel("Empty Password!", "The password cant be empty
please enter password")
        return
    elif(len(self.password.get())<5):
        messagebox.askokcancel("Weak Password!", "The length of the password
should be greater than or equal to 6")
        return

    #email validation
    regex='^[a-z0-9]+[\.\_]?[a-z0-9]+[@]\w+[.]\w{2,3}$'
    if not (re.search(regex,self.email.get())):
        messagebox.askokcancel("Invalid Email!", "The provided email is
invalid please try again")
        return

    #security Question validation
    if len(self.securtyAnswer.get())<1:
        messagebox.askokcancel("Empty Answer", "The answer cant be empty
please enter password")
        return

    conn=sqlite3.connect("pass.db")
    cur=conn.cursor()

    try:
        cur.execute("""CREATE TABLE pass(email VARCHAR(255),password
VARCHAR(255),SQ VARCHAR(255),SA VARCHAR(255))""")
    except:
        pass
    finally:
        enc1=Encryption(self.password.get(),self.email.get()).getEncryption()
```



```

        enc2=Encryption(self.securtyAnswer.get(),self.password.get()).getEncr
yption()
        val=(self.email.get(),enc1,self.clicked.get(),enc2)
        print(val)
        res=cur.execute("INSERT INTO pass VALUES(?,?,?,?)",val)
        conn.commit()
        conn=sqlite3.connect("password.db")
        cur=conn.cursor()
        try:
            cur.execute("""CREATE TABLE passtable(rID INTEGER PRIMARY
KEY,website VARCHAR(255),username VARCHAR(255),password VARCHAR(2000))""")
        except Exception as e:
            print("error creating passtable",e)
        finally:
            conn.commit()
        self.register.destroy()
        Login()

```

```
class Reset:
```

```

    def __init__(self) -> None:
        def loadLogin():
            self.reset.destroy()
            Login()
        self.reset=Tk()
        self.reset.title("Reset")
        self.reset["bg"]="#ff972e"
        self.reset.geometry("600x400")

        self.contents=Frame(self.reset)
        self.contents["bg"]="white"
        self.contents.pack(fill="both",expand="yes",padx=10,pady=38)

        conn=sqlite3.connect("pass.db")
        cur=conn.cursor()
        data=cur.execute("SELECT SQ from pass")
        for row in data:
            SQ=row[0]
        conn.close()

        #label
        LNewPassword=Label(self.contents,text="New Password",justify=LEFT)
        LQuestion=Label(self.contents,text="Question",justify=LEFT)
        Question=Label(self.contents,text=SQ,justify=LEFT)
        LAnswer=Label(self.contents,text="Answer",justify=LEFT)

```

```
#Input box
self.EPassword=Entry(self.contents)
self.EAnswer=Entry(self.contents)

#Button
verifyBtn=Button(self.contents,text="Change",command=self.resetPassword)
loginBtn=Button(self.reset,text="Login",command=loadLogin)

#all Placements
loginBtn.place(x=5,y=5,width=80,height=25)
LNewPassword.place(x=48,y=28,width=145,height=23,)
self.EPassword.place(x=208,y=25,width=222,height=32)
LQuestion.place(x=48,y=73,width=87,height=23,)
Question.place(x=208,y=74,width=300,height=27,)
LAnswer.place(x=48,y=116,width=74,height=23,)
self.EAnswer.place(x=208,y=113,width=222,height=32)
verifyBtn.place(x=227,y=153,width=84,height=37)

self.reset.mainloop()

def resetPassword(self):
    self.newPassword= self.EPassword.get()
    self.answer=self.EAnswer.get()
    self.encryptedText=""
    conn=sqlite3.connect("pass.db")
    cur=conn.cursor()

    try:
        record=cur.execute("""SELECT SA FROM pass""")
    except:
        print("error in database")
    finally:
        for row in record:
            self.encryptedText=row[0]
    password=Description(self.answer,self.encryptedText).getDecryption()
    global active_pass
    conn=sqlite3.connect("pass.db")
    cur=conn.cursor()
    storedData=cur.execute("select email,password from pass")
    storedEmail=""
    storedPassword=""
    for row in storedData:
        storedEmail=row[0]
        storedPassword=row[1]
```

```

if storedPassword == Encryption(password,storedEmail).getEncryption():
    active_pass=password

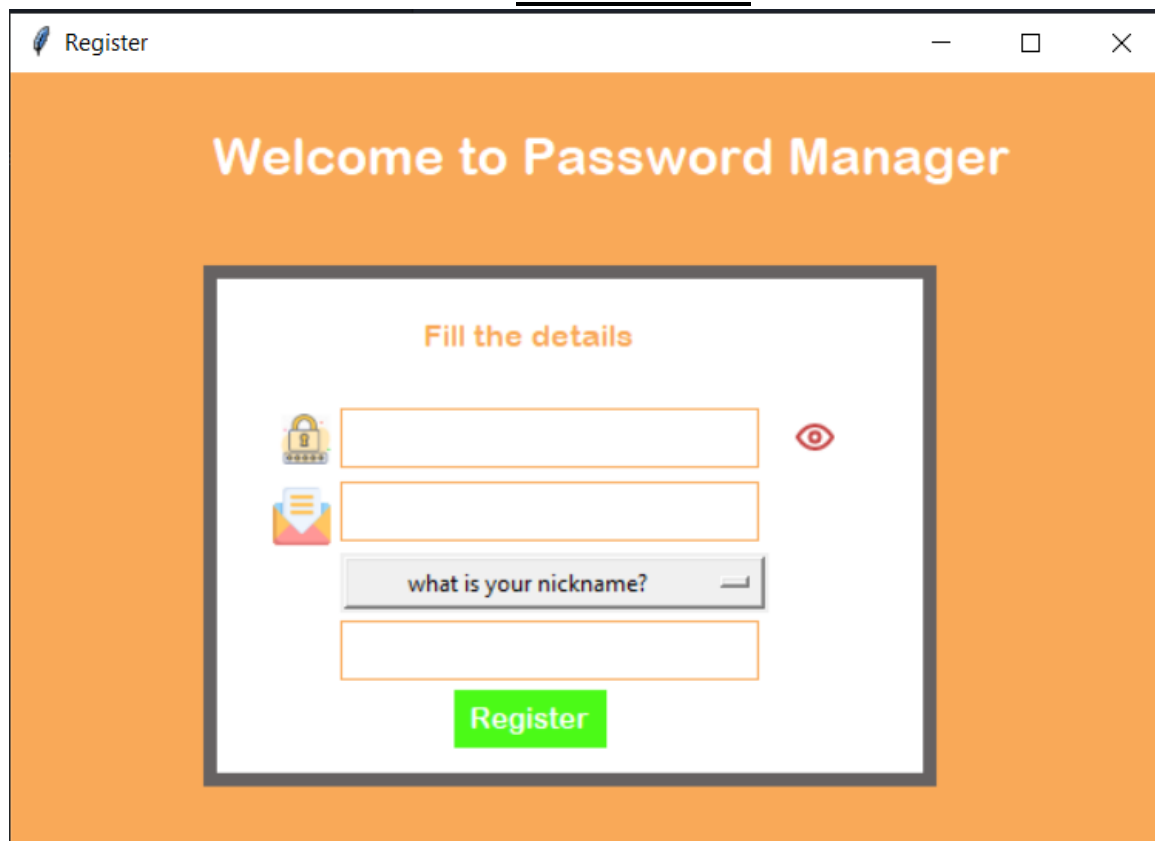
    enc1=Encryption(self.newPassword,storedEmail).getEncryption()
    enc2=Encryption(self.answer,self.newPassword).getEncryption()
    val=(enc1,enc2)
    cur.execute("UPDATE pass set password=?,SA=?",val)
    conn.commit()

    #changing encryption key of password stored in db
    conn=sqlite3.connect("password.db")
    cur=conn.cursor()
    data=cur.execute("SELECT rowid,password FROM passtable")
    for row in data:
        currentPass=Description(str(active_pass),str(row[1])).getDecrypti
on()
        encryptedPassWithNewKey=Encryption(self.newPassword,currentPass).
getEncryption()
        cur.execute("UPDATE passtable set
password='"+str(encryptedPassWithNewKey)+"' where rowid="+str(row[0])+"'")
        conn.commit()
        self.reset.destroy()

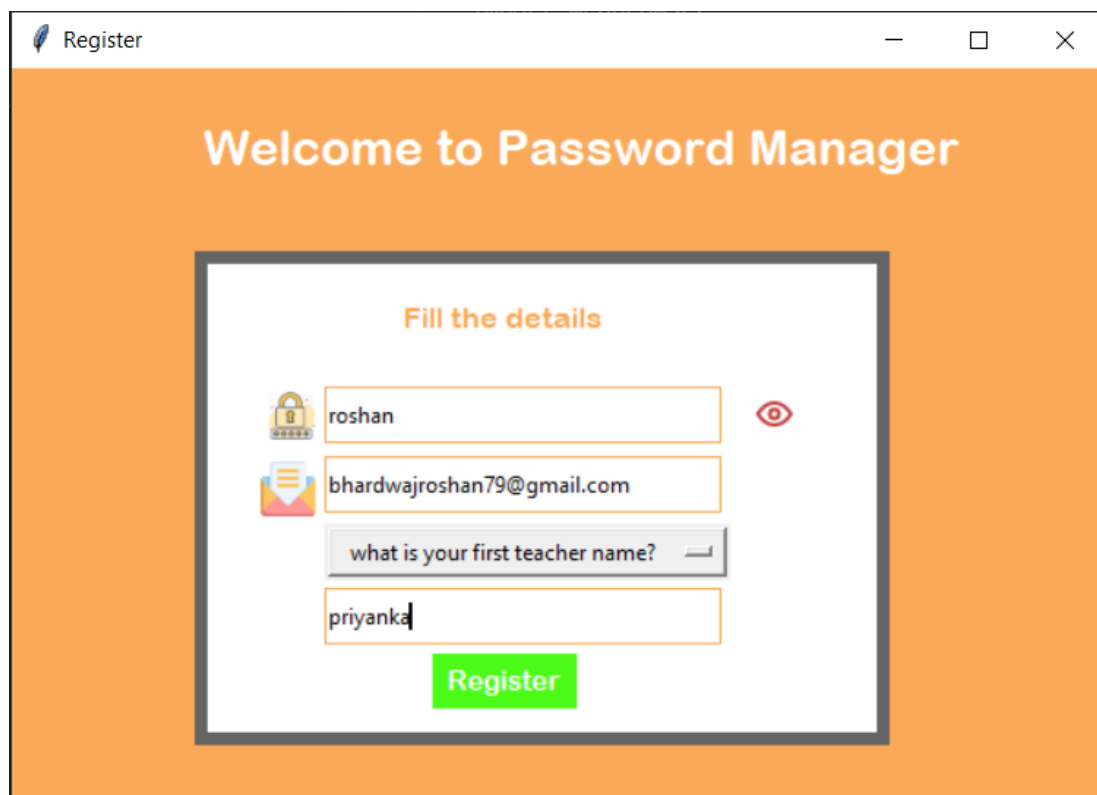
    Login()
else:
    messagebox("RESET FAILED","Please Check Your Security Answer")

if __name__=='__main__':
    conn=sqlite3.connect("pass.db")
    cur=conn.cursor()
    tables=cur.execute("SELECT name FROM sqlite_master WHERE type='table' AND
name='pass'").fetchall()
    if tables==[]:
        Register()
    else:
        Login()
    conn.commit()

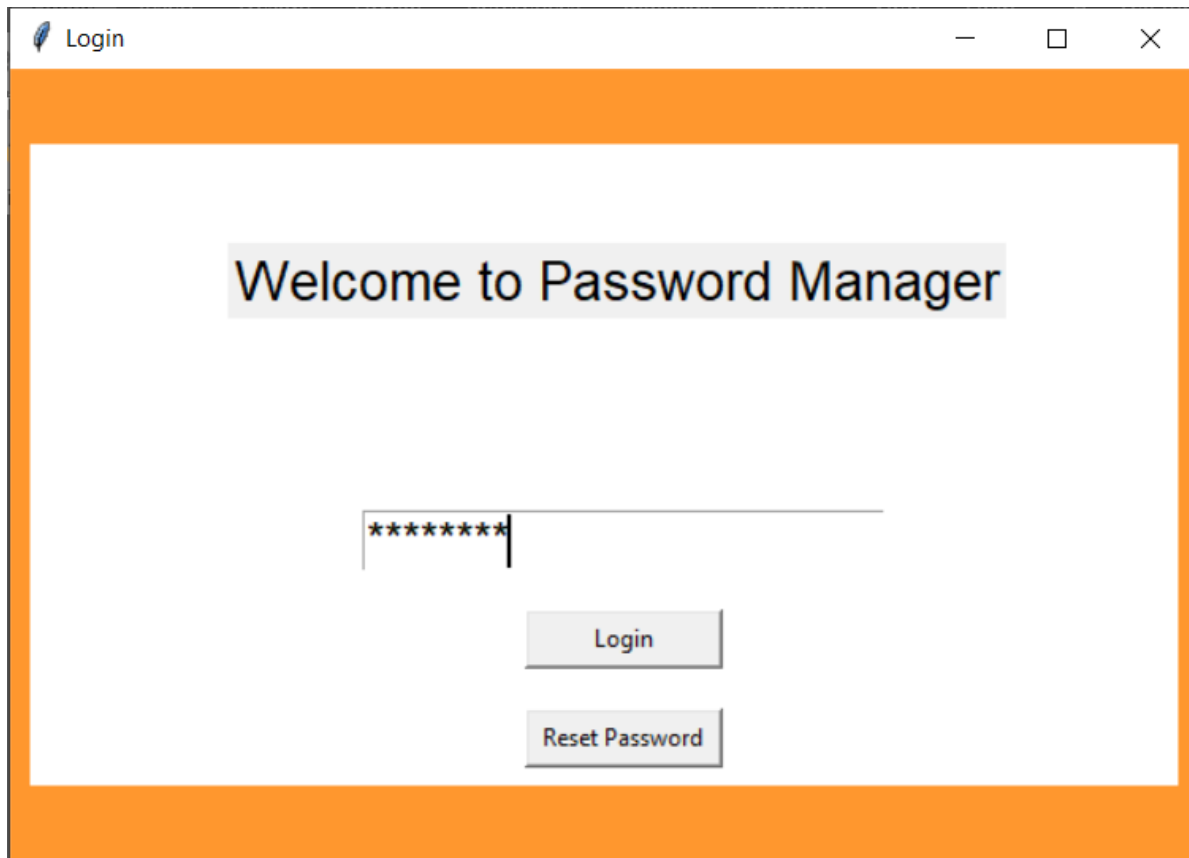
```

**SNAPSHOTS:**

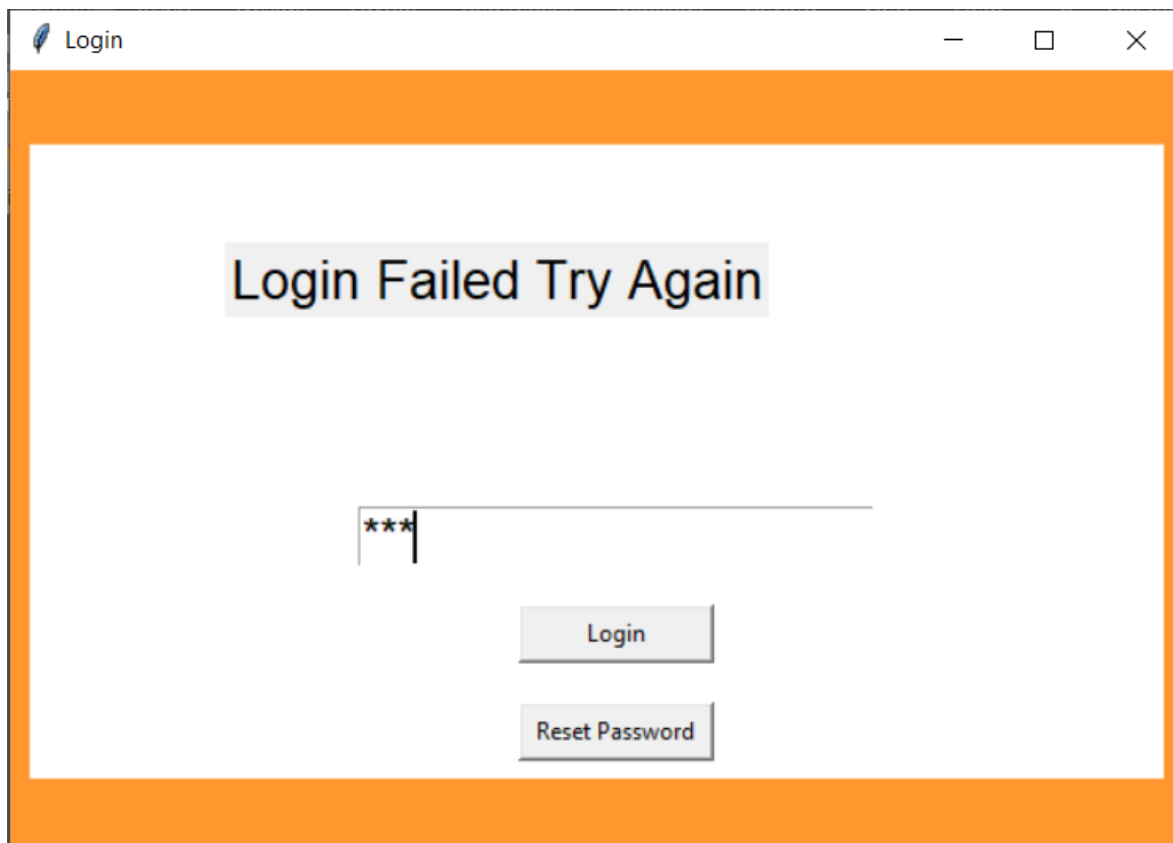
A screenshot of a web browser window titled "Register". The page has an orange background and displays the text "Welcome to Password Manager" in white. Below this, a white box with a grey border contains the heading "Fill the details" in orange. The form includes four input fields: a password field with a lock icon and a toggle eye icon, an email field with an envelope icon, a nickname dropdown menu with the text "what is your nickname?", and a final empty text field. A green "Register" button is positioned at the bottom of the form.



A second screenshot of the same "Register" page, but with sample data entered into the form fields. The password field contains "roshan", the email field contains "bhardwajroshan79@gmail.com", the nickname dropdown menu is set to "what is your first teacher name?", and the final text field contains "priyanka". The green "Register" button remains at the bottom.



A screenshot of a web application window titled "Login". The window has a white background with an orange border. In the center, the text "Welcome to Password Manager" is displayed in a large, black, sans-serif font. Below the text, there is a password input field containing eight asterisks "\*\*\*\*\*". Underneath the input field, there are two buttons: "Login" and "Reset Password", both with a light gray background and a thin black border.



A screenshot of the same "Login" window, but with a different message. The text "Login Failed Try Again" is displayed in the center in a large, black, sans-serif font. Below the text, the password input field now contains only three asterisks "\*\*\*". The "Login" and "Reset Password" buttons remain below the input field.

Accounts

Account List

No	Website	Username	Password
----	---------	----------	----------

search

search

Account Data

Website

Username

Password


Account Data

Website

Username

Password

Successfull

 New Record Created Successfully

Accounts

Account List

No	Website	Username	Password
1	www.google.com	roshanb79	11100101111100011111001011

search






search

Account Data

Website

Username

Password

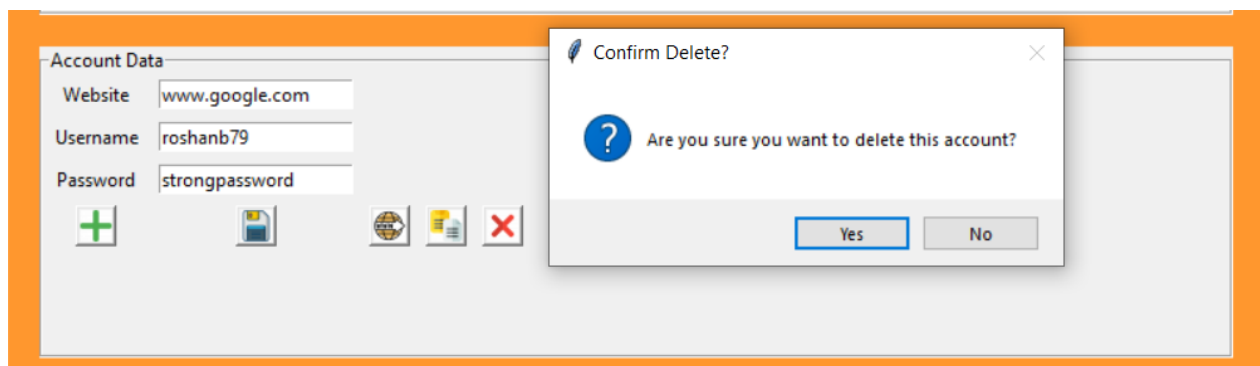
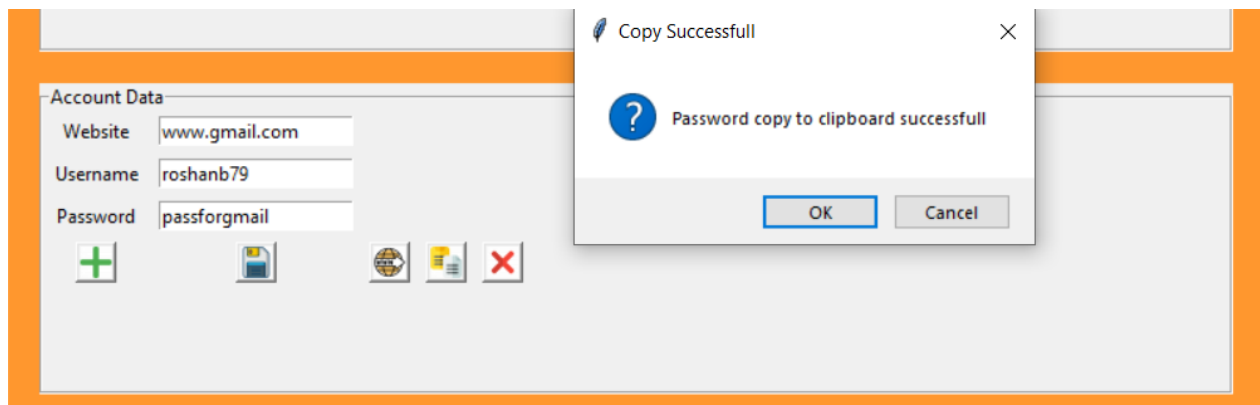
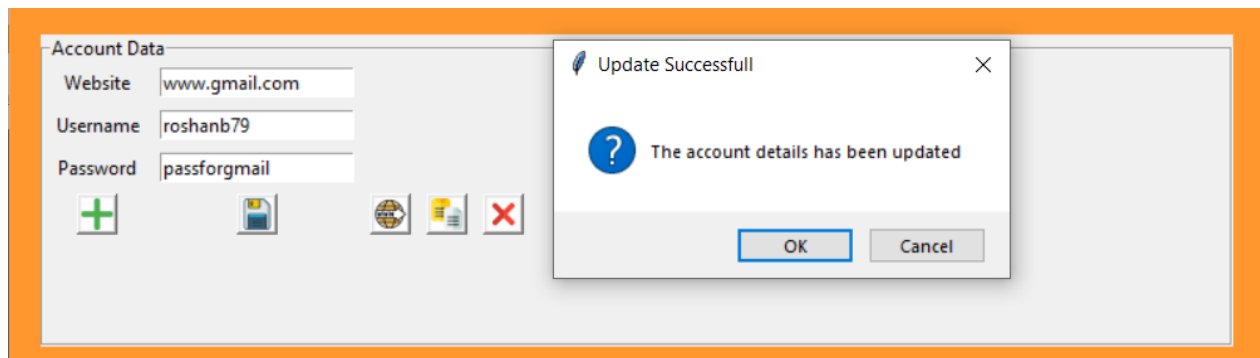
Accounts

Account List

No	Website	Username	Password
8	www.gmail.com	roshanb79	11100101111100011111001011

search

search





Reset

Login

New Password

satish

Question

what is your first teacher name?

Answer

priyanka

Change

## **FUTURE SCOPE**

- 1) Make browser extension so that it can auto fill the username and password in browser.
- 2) Make it online so that it can share username and password with other devices and even on mobile phones which can make it cross-platform compatible.
- 3) Improve Encryption algorithm to make it strong.
- 4) Implement other better way to reset password.

## **LIMITATIONS:**

It resets the password with security answer it is easy for attacker to obtain the answer of security question if attacker is friend of user.

User has to manually type the username in browser and paste the password.

If “password.db” file is deleted all the password will be lost.

## **REFERENCES:**

- [1] <https://youtu.be/i4qLI9Imkqw>
- [2] <https://docs.python.org/3/library/tkinter.html>
- [3] <https://www.geeksforgeeks.org/python-gui-tkinter/>