Certificate

This is to certify that student Lakshay Bhushan of class XII-A has successfully completed the Computer Science project "Python – MySQL Password Manager" under the guidance of Mr. Amar Sir (subject teacher) during the academic session 2020-2021.

Teacher's Sign. *Principal's Sign*

Table Of Contents

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Mr Amar sir as well as our principal Miss Anjali Atri Mam who gave me the golden opportunity to do this wonderful project on the topic " Python – MySQL Password Manager", 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.

Thank you

Introduction

The Python – MySQL Password Manager is an application built using the famous Python programming language at frontend & at the backend Oracle's MySQL with the GUI.

This project allows a user to safely secure his/her 's highly crucial passwords and enables to view them anytime - anywhere on their respective Personal Computers. Some salient features of this application are:

1. **Master Password** : This feature allows the user to create a master password to log on the application.
2. **Use of Asterisk** : Hides the password entered by the user.
3. **Show all passwords** : Allows a user to view all the passwords in a table.
4. **Log Off Button :** This function button enables a user to exit the current window.
5. **Message Window :** This function enables a user to know that his/her 's details have been submitted or not or to show any wrong entry in a text field.

System Requirements

MINIMUM REQUIREMENTS:

- Processors: Intel Atom® processor or Intel® Core™ i3 processor.
- Disk Space: 1 GB.
- Operating Systems: Windows10 , Mac OS, and Linux
- Python* versions: 2.7.X , 3.8.X , etc...

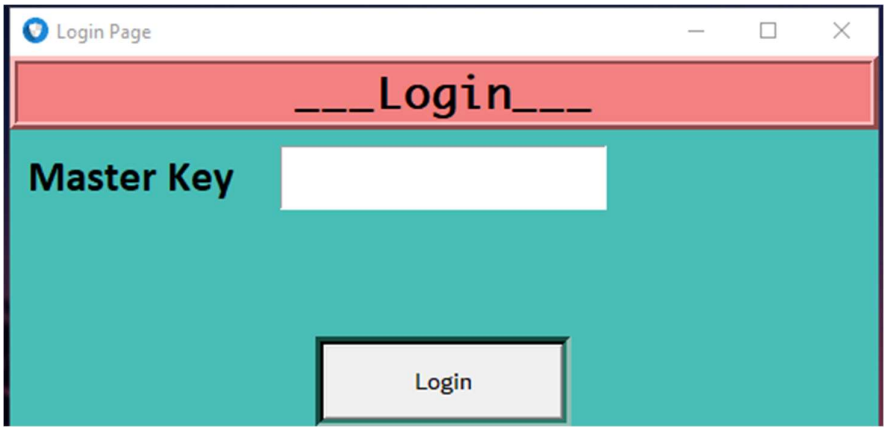
SOFTWARES REQUIRED :

- Python 3.9.x

- MySQL 8.0.xx

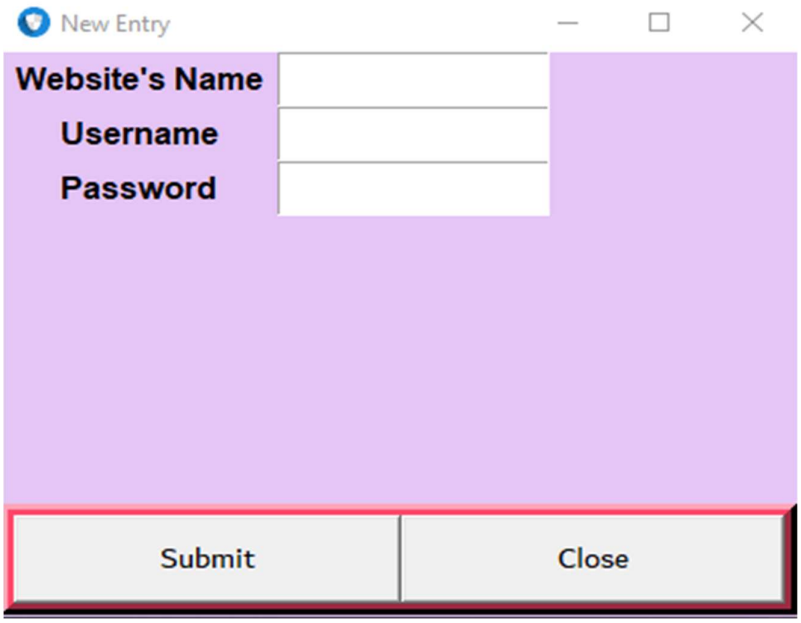
Project at a glance

Login Window

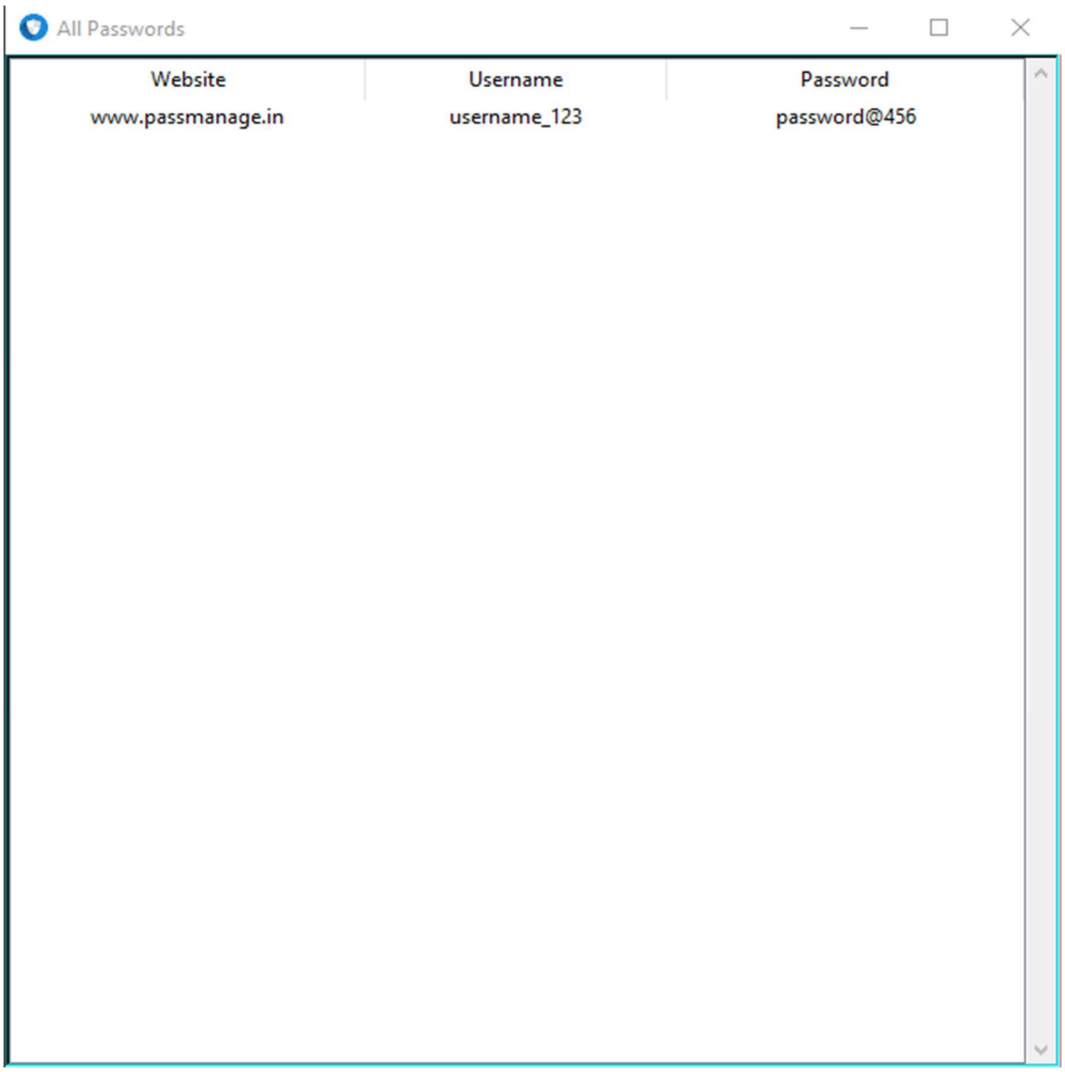


The Main Window:

New Entry Window:



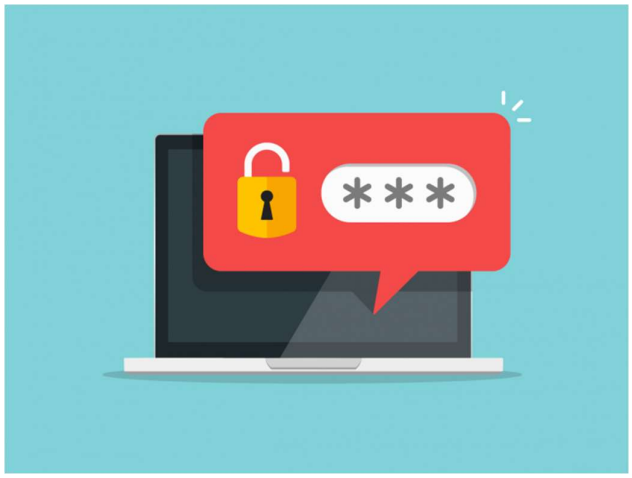
View Password window:



MySQL Window at the backend:

```
Select MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use password_manager
Database changed
mysql> select * from DATB;
+-------------------+--------------+--------------+
| Website           | Username     | Password     |
+-------------------+--------------+--------------+
| www.passmanage.in | username_123 | password@456 |
+-------------------+--------------+--------------+
1 row in set (0.00 sec)

mysql>
```

- Some Important Images (.Png file , .ico files) used in the project are listed below:



← Main Python file image file

Icon image used in the program

Note: All these images used in the applications are either copyright free or are licensable or belongs to creative commons. cc

Project Code

Main.py

```
#importing modules
from tkinter import *
from tkinter import ttk
import mysql.connector
from tkinter import messagebox

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="poah",
    database="password_manager"
)

login=Tk()
login.geometry('543x232+50+100')
login.maxsize(543,232)
login.minsize(532,232)
login.iconbitmap("iconpm.ico")
login.title('Login Page')
login.configure(bg='#49beb7')


def login_command():
    if mainpage_entry.get() == "poah" :  #Set your
password/masterkey here --------
        mainpage_entry.delete(0,'end')
        return mainwindow()
        print(mainwindow())
    else:
        messagebox.showerror("Please Try Again","Access
Denied!")
```

```
Label(login, text ="___Login___",font = "Monaco 20
bold",relief="ridge",bg="#f38181",
borderwidth=5).pack(fill = X)
Label(login, text="Master Key", font="Calibri 20 bold" ,bg
='#49beb7').place(x=8,y=55)
login_pass = StringVar()
mainpage_entry = Entry(login,
textvariable=login_pass,show="*",font = "monaco 10 ")
mainpage_entry.pack(side ='top',ipadx=20 , ipady=10
,pady=10)
Login_button=Frame(login, bg="#247e6c", relief="sunken",
borderwidth=5)
Login_button.pack(side="bottom",anchor="s")
Button(Login_button, text="Login",font="Dubai",command =
login_command).grid(ipadx =50)

def mainwindow():
    root = Tk()
    root.geometry("570x532+630+220")
    root.iconbitmap("iconpm.ico")
    root.minsize(570, 532)
    root.maxsize(570, 532)

    root.title("Password Manager")

    def add():
        def add_entry():
            mycur = mydb.cursor()
            command = f"insert into datb
values('{websiteentry.get()}', '{usernameentry.get()}',
'{passwordentry.get()}')"
            mycur.execute(command)
            messagebox.showinfo("Password Manager", "Your
details have been submitted!")

            mydb.commit()

            websiteentry.delete(0, 'end')
            usernameentry.delete(0, 'end')
```

```
            passwordentry.delete(0, 'end')

        def close():
            addwin.destroy()

        addwin = Tk()
        addwin.geometry("393x300+200+400")
        addwin.maxsize(393, 300)
        addwin.minsize(393, 300)
        addwin.configure(bg='#E5C4F6')
        addwin.iconbitmap("iconpm.ico")
        addwin.title("New Entry")
        Label(addwin, text=" Website's Name ",
font="Helvetica 12 bold", bg="#E5C4F6").grid(row=0)
        Label(addwin, text="Username", font="Helvetica 12
bold", bg="#E5C4F6").grid(row=1)
        Label(addwin, text="Password", font="Helvetica 12
bold", bg="#E5C4F6").grid(row=2)

        Websiteval = StringVar()
        Usernameval = StringVar()
        Passwordval = StringVar()

        websiteentry = Entry(addwin,
textvariable=Websiteval)
        usernameentry = Entry(addwin,
textvariable=Usernameval)
        passwordentry = Entry(addwin,
textvariable=Passwordval)

        websiteentry.grid(row=0, column=1, ipady=5,
ipadx=5)
        usernameentry.grid(row=1, column=1, ipady=5,
ipadx=5)
        passwordentry.grid(row=2, column=1, ipady=5,
ipadx=5)
        button_frame = Frame(addwin, bg="#ff496c",
relief="raised", borderwidth=6)
        button_frame.place(y=240)
        Button(button_frame, text="Submit", font="Dubai",
command=add_entry).grid(row=0, ipadx=65)
```

```
        Button(button_frame, text="Close", font="Dubai",
command=close).grid(row=0, column=1, ipadx=70)


        addwin.mainloop()

    def logoff():
        root.destroy()

#Defining previous data
    def prevdata():
        prev = Tk()
        prev.geometry("595x600+1250+100")
        prev.maxsize(595, 600)
        prev.minsize(595, 600)
        prev.configure(bg='#f08080')
        prev.iconbitmap("iconpm.ico")
        prev.title("All Passwords")

        frame_data = Frame(prev, bg="#11cbd7",
borderwidth=2, relief="sunken")
        frame_data.pack()

        # treeview table definition

        tree_scrollbar = Scrollbar(frame_data)
        tree_scrollbar.pack(side=RIGHT, fill=Y)
        my_tree = ttk.Treeview(frame_data,
yscrollcommand=tree_scrollbar.set)
        my_tree.pack(ipady=200)

        tree_scrollbar.config(command=my_tree.yview)

        my_tree['columns'] = ("Website", "Username",
"Password")

        my_tree.column("#0", width=0, stretch=NO)
        my_tree.column("Website", anchor="center",
width=200)
        my_tree.column("Username", anchor=CENTER,
width=170)
```

```
        my_tree.column("Password", anchor=CENTER,
width=200)

        my_tree.heading("#0", text="", anchor="w")
        my_tree.heading("Website", text="Website",
anchor="center")
        my_tree.heading("Username", text="Username",
anchor="center")
        my_tree.heading("Password", text="Password",
anchor="center")


        mycur = mydb.cursor()
        mycur.execute("SELECT * FROM DATB")
        result = mycur.fetchall()

        count = 0
        for rec in result:
            my_tree.insert(parent='', index='end',
iid=count, text="", values=(rec[0], rec[1], rec[2]))
            count += 1

        prev.mainloop()

    # main loop code
    frame_label = Frame(root, bg="#40a6c3",
relief="groove", borderwidth=5)
    frame_label.pack(fill=X)
    Label(frame_label, text="Password Manager",
font="Monaco 20 bold").pack()

    frame_button = Frame(root, bg="#40a6c3",
relief="groove", borderwidth=5)
    frame_button.pack(side="bottom", anchor="s", fill=X)
    Button(frame_button, text="Add New", font="Monaco 15",
command=add).grid(ipadx =20)
    Button(frame_button ,text="Show All
Passwords",font="Monaco 15" ,command=
prevdata).grid(row=0, column=1)
    Button(frame_button, text="Log Off", font="Monaco 15",
command=logoff).grid(row=0, column=2 ,ipadx=40)
```

```
    # Image Source
    photo = PhotoImage(file="password.png",master=root)
    bg =Label(root,image= photo)
    bg.image=photo
    bg.pack(padx=20,pady=20)

login.mainloop()
```

Some Explanation Of The Written Code:

1. **Use of Python Modules :** To simply the code or to get rid of repeating the same lines of code by 1000 times, I have imported python's integrated modules namely tkinter , mysql.connector (to connect MySQL with python).

```
#importing modules
from tkinter import *
from tkinter import ttk
import mysql.connector
from tkinter import messagebox
```

2. **Establishing connection between MySQL and Python :** To establish connection , the small piece of code written here :➢

```
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="poah", database="password_manager")
```

3. Login Window code with the help of GUI :

```
login=Tk()
login.geometry('543x232+50+100') # Geometry of window
login.maxsize(543,232) #Max and Min Size of window
login.minsize(532,232)
login.iconbitmap("iconpm.ico") # Use of icon
login.title('Login Page') # The title name
login.configure(bg='#49beb7') # Background colour
```

4. Basic Login window structure with use of Buttons and framing , defining Borders , Anchors and Side.

```
Label(login, text ="___Login___",font = "Monaco 20
bold",relief="ridge",bg="#f38181",
borderwidth=5).pack(fill = X)
Label(login, text="Master Key", font="Calibri 20 bold"
,bg ='#49beb7').place(x=8,y=55)
login_pass = StringVar()
mainpage_entry = Entry(login,
textvariable=login_pass,show="*",font = "monaco 10 ")
mainpage_entry.pack(side ='top',ipadx=20 , ipady=10
,pady=10)
Login_button=Frame(login, bg="#247e6c", relief="sunken",
borderwidth=5)
Login_button.pack(side="bottom",anchor="s")
Button(Login_button, text="Login",font="Dubai",command =
login_command).grid(ipadx =50)
```

5. Defining function for Login window and error Message box :

```
def login_command():
    if mainpage_entry.get() == "poah" :  #Set your
password/masterkey here --------
        mainpage_entry.delete(0,'end')
```

```
        return mainwindow()
        print(mainwindow())

    else:
        messagebox.showerror("Please Try Again","Access
Denied!")
```

6. Defining a function - Main Window and inside it I have defined another two functions for separate windows, its size , geometry ,etc.

```
def mainwindow():
    root = Tk()
    root.geometry("570x532+630+220")
    root.iconbitmap("iconpm.ico")
    root.minsize(570, 532)
    root.maxsize(570, 532)

    root.title("Password Manager")

    def add():
        def add_entry():
            mycur = mydb.cursor()
            command = f"insert into datb
values('{websiteentry.get()}', '{usernameentry.get()}',
'{passwordentry.get()}')"
            mycur.execute(command)
            messagebox.showinfo("Password Manager",
"Your details have been submitted!")

            mydb.commit()

            websiteentry.delete(0, 'end')
            usernameentry.delete(0, 'end')
            passwordentry.delete(0, 'end')

        def close():
            addwin.destroy()
```

```
        addwin = Tk()
        addwin.geometry("393x300+200+400")
        addwin.maxsize(393, 300)
        addwin.minsize(393, 300)
        addwin.configure(bg='#E5C4F6')
        addwin.iconbitmap("iconpm.ico")
        addwin.title("New Entry")
        Label(addwin, text=" Website's Name ",
font="Helvetica 12 bold", bg="#E5C4F6").grid(row=0)
        Label(addwin, text="Username", font="Helvetica
12 bold", bg="#E5C4F6").grid(row=1)
        Label(addwin, text="Password", font="Helvetica
12 bold", bg="#E5C4F6").grid(row=2)

        Websiteval = StringVar()
        Usernameval = StringVar()
        Passwordval = StringVar()

        websiteentry = Entry(addwin,
textvariable=Websiteval)
        usernameentry = Entry(addwin,
textvariable=Usernameval)
        passwordentry = Entry(addwin,
textvariable=Passwordval)

        websiteentry.grid(row=0, column=1, ipady=5,
ipadx=5)
        usernameentry.grid(row=1, column=1, ipady=5,
ipadx=5)
        passwordentry.grid(row=2, column=1, ipady=5,
ipadx=5)
        button_frame = Frame(addwin, bg="#ff496c",
relief="raised", borderwidth=6)
        button_frame.place(y=240)
        Button(button_frame, text="Submit",
font="Dubai", command=add_entry).grid(row=0, ipadx=65)
        Button(button_frame, text="Close",
font="Dubai", command=close).grid(row=0, column=1,
ipadx=70)
```

```
        addwin.mainloop()

    def logoff():
        root.destroy()

#Defining previous data
    def prevdata():
        prev = Tk()
        prev.geometry("595x600+1250+100")
        prev.maxsize(595, 600)
        prev.minsize(595, 600)
        prev.configure(bg='#f08080')
        prev.iconbitmap("iconpm.ico")
        prev.title("All Passwords")

        frame_data = Frame(prev, bg="#11cbd7",
borderwidth=2, relief="sunken")
        frame_data.pack()

        # treeview table definition

        tree_scrollbar = Scrollbar(frame_data)
        tree_scrollbar.pack(side=RIGHT, fill=Y)
        my_tree = ttk.Treeview(frame_data,
yscrollcommand=tree_scrollbar.set)
        my_tree.pack(ipady=200)

        tree_scrollbar.config(command=my_tree.yview)

        my_tree['columns'] = ("Website", "Username",
"Password")

        my_tree.column("#0", width=0, stretch=NO)
        my_tree.column("Website", anchor="center",
width=200)
        my_tree.column("Username", anchor=CENTER,
width=170)
        my_tree.column("Password", anchor=CENTER,
width=200)
```

```
        my_tree.heading("#0", text="", anchor="w")
        my_tree.heading("Website", text="Website",
anchor="center")
        my_tree.heading("Username", text="Username",
anchor="center")
        my_tree.heading("Password", text="Password",
anchor="center")


        mycur = mydb.cursor()
        mycur.execute("SELECT * FROM DATB")
        result = mycur.fetchall()

        count = 0
        for rec in result:
            my_tree.insert(parent='', index='end',
iid=count, text="", values=(rec[0], rec[1], rec[2]))
            count += 1

        prev.mainloop()

    # main loop code
    frame_label = Frame(root, bg="#40a6c3",
relief="groove", borderwidth=5)
    frame_label.pack(fill=X)
    Label(frame_label, text="Password Manager",
font="Monaco 20 bold").pack()

    frame_button = Frame(root, bg="#40a6c3",
relief="groove", borderwidth=5)
    frame_button.pack(side="bottom", anchor="s",
fill=X)
    Button(frame_button, text="Add New", font="Monaco
15", command=add).grid(ipadx =20)
    Button(frame_button ,text="Show All
Passwords",font="Monaco 15" ,command=
prevdata).grid(row=0, column=1)
    Button(frame_button, text="Log Off", font="Monaco
15", command=logoff).grid(row=0, column=2 ,ipadx=40)
login.mainloop()
```

7. The Logoff Command :

```
def close():
    addwin.destroy()
```

8. Defining a function / Creating a window where user can enter his/her ‘s details (Safely):

```
 def add():
    def add_entry():
        mycur = mydb.cursor()
        command = f"insert into datb
values('{websiteentry.get()}', '{usernameentry.get()}',
'{passwordentry.get()}')"
        mycur.execute(command)
        messagebox.showinfo("Password Manager", "Your
details have been submitted!")

        mydb.commit()

        websiteentry.delete(0, 'end')
        usernameentry.delete(0, 'end')
        passwordentry.delete(0, 'end')

    def close():
        addwin.destroy()

    addwin = Tk()
    addwin.geometry("393x300+200+400")
    addwin.maxsize(393, 300)
    addwin.minsize(393, 300)
    addwin.configure(bg='#E5C4F6')
    addwin.iconbitmap("iconpm.ico")
    addwin.title("New Entry")
    Label(addwin, text=" Website's Name ",
font="Helvetica 12 bold", bg="#E5C4F6").grid(row=0)
    Label(addwin, text="Username", font="Helvetica 12
bold", bg="#E5C4F6").grid(row=1)
    Label(addwin, text="Password", font="Helvetica 12
```

```
bold", bg="#E5C4F6").grid(row=2)

    Websiteval = StringVar()
    Usernameval = StringVar()
    Passwordval = StringVar()
#The Entry widgets
    websiteentry = Entry(addwin,
textvariable=Websiteval)
    usernameentry = Entry(addwin,
textvariable=Usernameval)
    passwordentry = Entry(addwin,
textvariable=Passwordval)

    websiteentry.grid(row=0, column=1, ipady=5,
ipadx=5)
    usernameentry.grid(row=1, column=1, ipady=5,
ipadx=5)
    passwordentry.grid(row=2, column=1, ipady=5,
ipadx=5)
    button_frame = Frame(addwin, bg="#ff496c",
relief="raised", borderwidth=6)
    button_frame.place(y=240)
    Button(button_frame, text="Submit", font="Dubai",
command=add_entry).grid(row=0, ipadx=65)
    Button(button_frame, text="Close", font="Dubai",
command=close).grid(row=0, column=1, ipadx=70)
    addwin.mainloop()
```

9. Defining a Function / Creating a window where user can view the saved password using treeview.

```
#Defining previous data
    def prevdata():
        prev = Tk()
        prev.geometry("595x600+1250+100")
        prev.maxsize(595, 600)
        prev.minsize(595, 600)
        prev.configure(bg='#f08080')
        prev.iconbitmap("iconpm.ico")
```

```
        prev.title("All Passwords")

        frame_data = Frame(prev, bg="#11cbd7",
borderwidth=2, relief="sunken")
        frame_data.pack()

        # treeview table definition

        tree_scrollbar = Scrollbar(frame_data)
        tree_scrollbar.pack(side=RIGHT, fill=Y)
        my_tree = ttk.Treeview(frame_data,
yscrollcommand=tree_scrollbar.set)
        my_tree.pack(ipady=200)

        tree_scrollbar.config(command=my_tree.yview)

        my_tree['columns'] = ("Website", "Username",
"Password")

        my_tree.column("#0", width=0, stretch=NO)
        my_tree.column("Website", anchor="center",
width=200)
        my_tree.column("Username", anchor=CENTER,
width=170)
        my_tree.column("Password", anchor=CENTER,
width=200)

        my_tree.heading("#0", text="", anchor="w")
        my_tree.heading("Website", text="Website",
anchor="center")
        my_tree.heading("Username", text="Username",
anchor="center")
        my_tree.heading("Password", text="Password",
anchor="center")


        mycur = mydb.cursor()
        mycur.execute("SELECT * FROM DATB")
        result = mycur.fetchall()

        count = 0
```

```
        for rec in result:
            my_tree.insert(parent='', index='end',
iid=count, text="", values=(rec[0], rec[1], rec[2]))
            count += 1

        prev.mainloop()
```

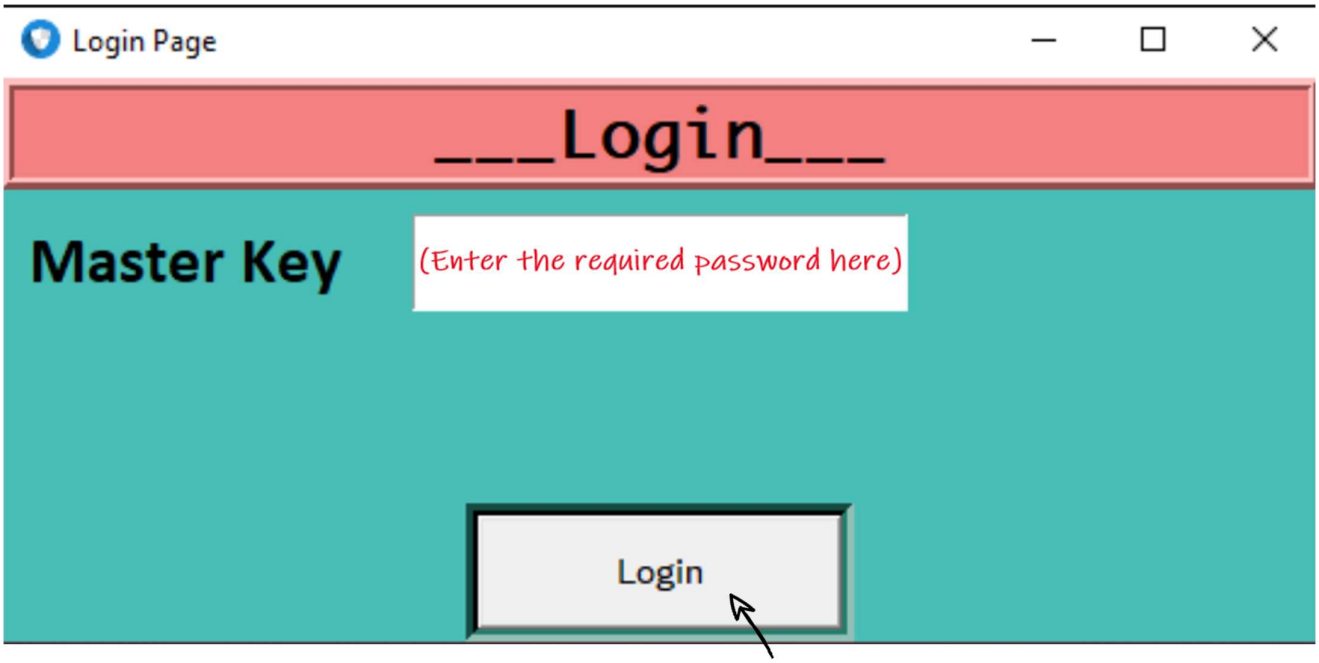
10. Importing Image in the tkinter window (Main Window) from my PC :

```
# Image Source
photo = PhotoImage(file="password.png",master=root)
bg =Label(root,image= photo)
bg.image=photo
bg.pack(padx=20,pady=20)
```

This Image

Output / Result

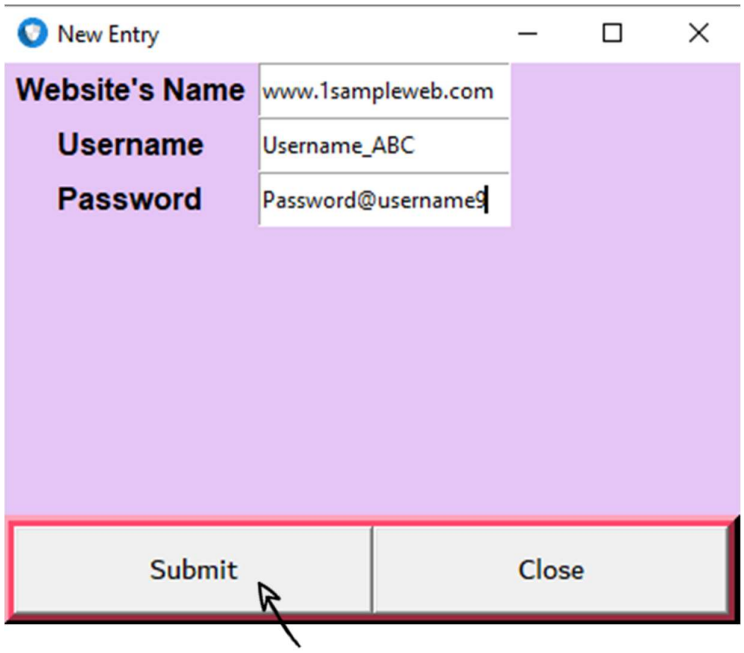
➢Enter the Master key here :



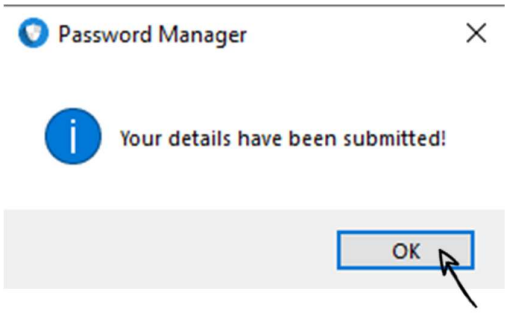
Then click on ‘Login’ button.

➢This will open the main window :

➢ Click on ‘Add New’ button to add new passwords :

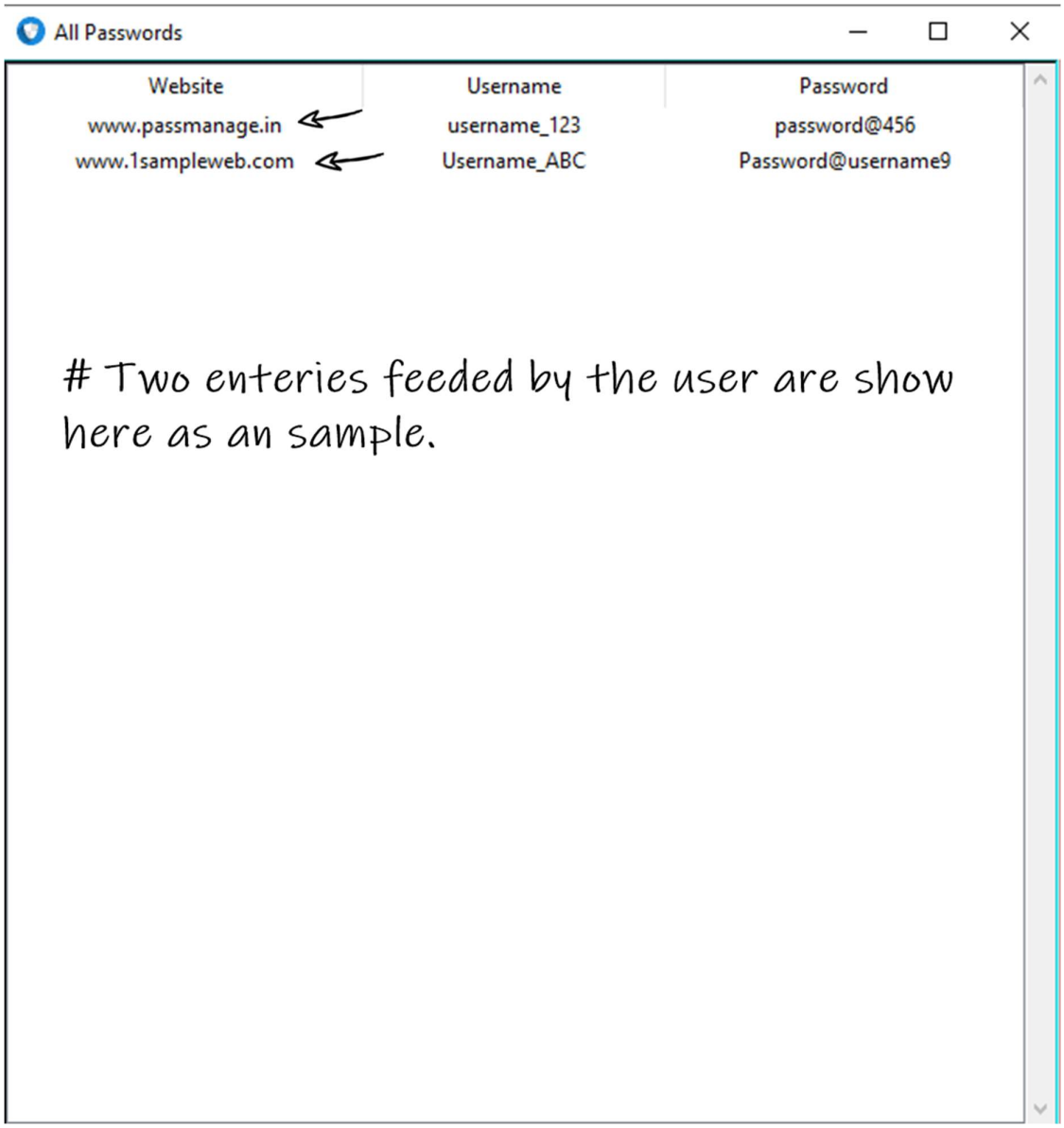


Then click on ‘Submit’ button. Then a message will pop up.



Click on OK button to continue

➢ To show all the saved passwords , on the main window click on ‘Show All Passwords’. This button will show up a window where the person can see all his/her ‘s saved passwords.


All Passwords
Website
Username
Password
www.passmanage.in
username_123
password@456
www.1sampleweb.com
Username_ABC
Password@username9
Two enteries feeded by the user are show here as an sample.

Conclusion

- In today's world of frequent cyber crime like phishing , this application can be used to prevent such privacy losses.
- This Python – MySQL Password works offline, hence, it doesn't require a high Speed internet connection.
- Now a user doesn't need to remember huge lengths of passwords. He /She can store infinitely many passwords in the application's database.

Maintenance:

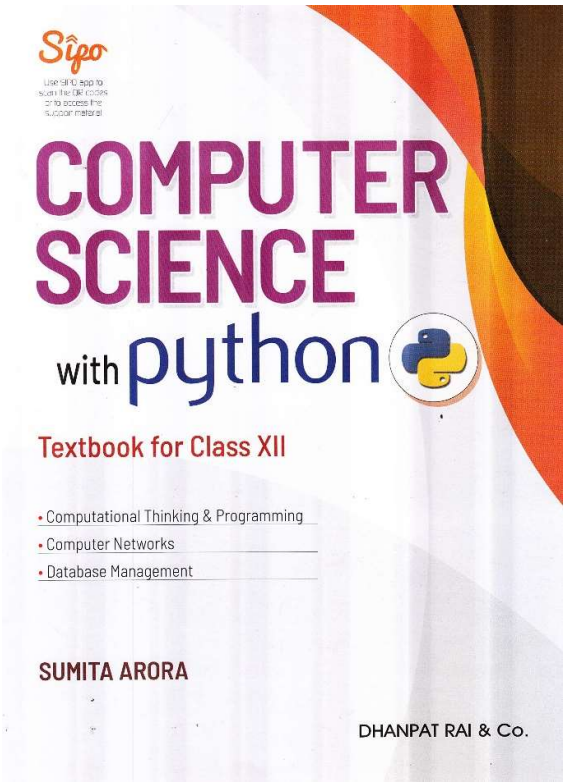
- This application by default has many user's wanted features. A python programmer can edit its source code which is available on GitHub so that he/she can add new features to this application.

https://github.com/lakshaybhushan/Password-Manager

Note: The Word GitHub in the **Maintenance paragraph** is a Hyperlink that redirects the user to the GitHub's webpage to edit the source code there.

Bibliography

- **Websites :**
  - https://stackoverflow.com/
  - https://www.w3schools.com/
  - https://docs.python.org/3/
  - https://www.programiz.com/
- **Books:**
  - **Computer Science with Python Class XII by Sumita Arora.**



stackoverflow

w3schools.com

python™

programiz