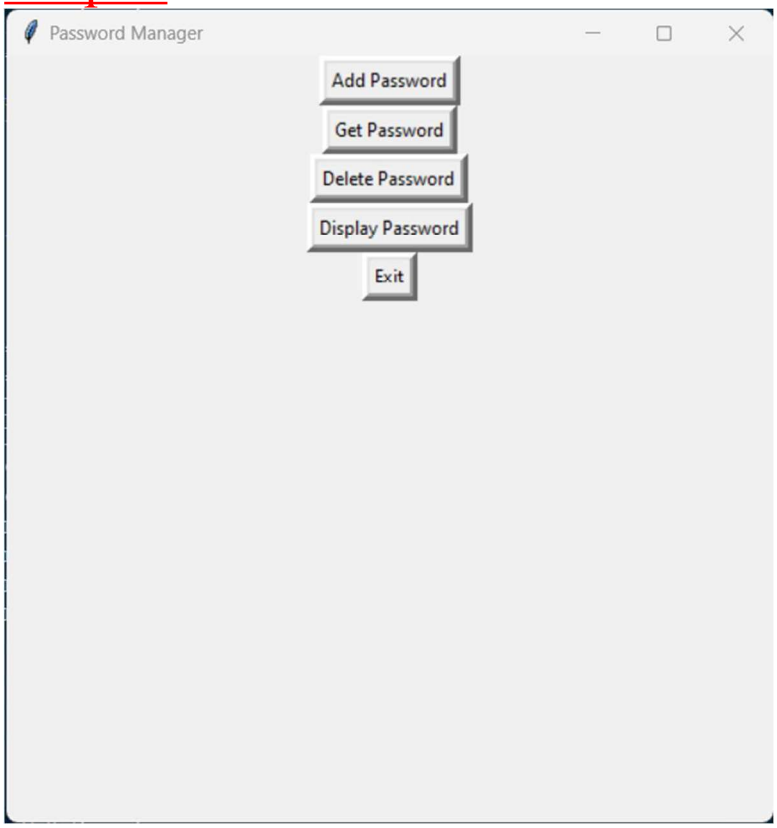
Code:

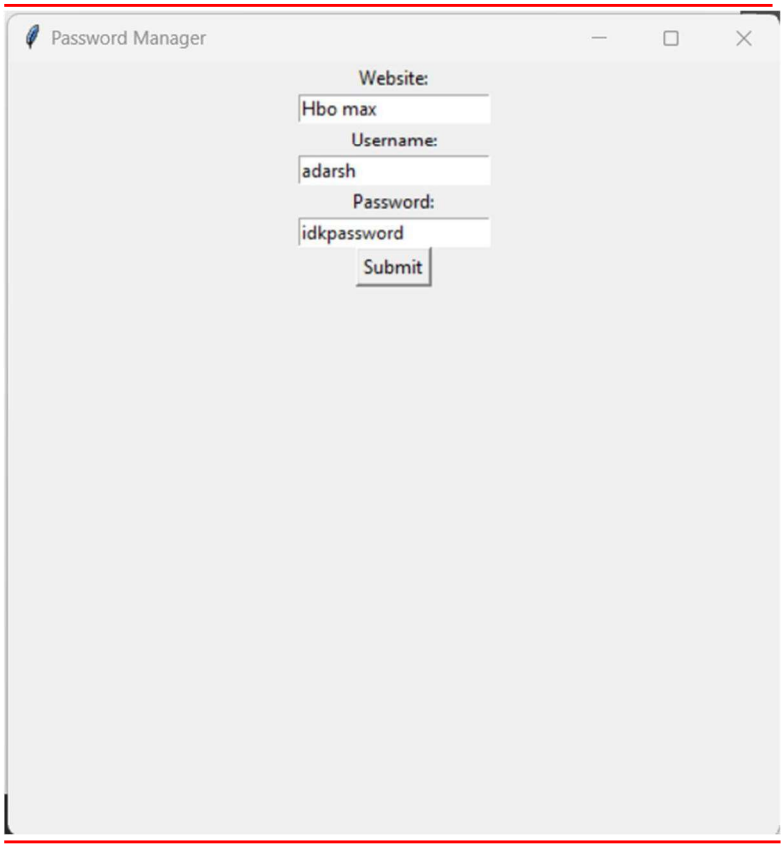
```
import os
import tkinter as tk
from tkinter import *
fileloc="D:\All coding\Python\passwords.txt"
def add_password(website, username, password):
    with open(fileloc, "a") as f:
        f.write(f"{website} | {username} | {password}\n")
        print(f"Password for {website} added successfully!")

def adpass():
    ovr = tk.Toplevel()
    ovr.geometry('500x500')
    v = StringVar()
    v2 = StringVar()
    v1 = StringVar()
    uweb = tk.Label(ovr, text="Website:")
    uweb.pack()
    upweb3 = tk.Entry(ovr, textvariable=v)
    upweb3.pack()
    uname = tk.Label(ovr, text="Username:")
    uname.pack()
    uname2 = tk.Entry(ovr, textvariable=v2)
    uname2.pack()
    upassword = tk.Label(ovr, text="Password:")
    upassword.pack()
    upasswordn = tk.Entry(ovr, textvariable=v1)
    upasswordn.pack()
    def submit():
        website = v.get()
        username = v2.get()
        password = v1.get()
        add_password(website, username, password)
    submit_button = tk.Button(ovr, text="Submit", command=submit)
    submit_button.pack()


def get_password(website):
    with open(fileloc, "r") as f:
        for line in f:
            if website in line:
                return line.split("|")[2].strip()
```

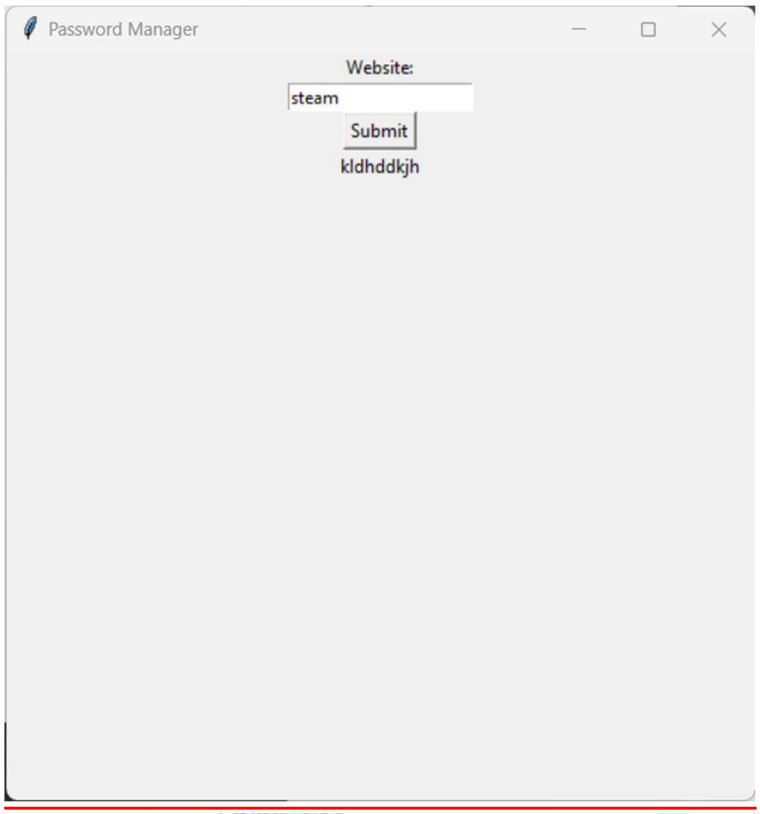
```
        return "Password not found."

def getpass():
    ovr = tk.Toplevel()
    ovr.geometry('500x500')
    v = StringVar()
    uweb = tk.Label(ovr, text="Website:")
    uweb.pack()
    upweb3 = tk.Entry(ovr, textvariable=v)
    upweb3.pack()
    def submit():
        website = v.get()
        tk.Label(ovr,text=get_password(website)).pack()
    submit_button = tk.Button(ovr, text="Submit", command=submit)
    submit_button.pack()

def delete_password(website):
    with open(fileloc, "r") as f:
        lines = f.readlines()
    with open(fileloc, "w") as f:
        for line in lines:
            if website not in line:
                f.write(line)
    print(f'Password for {website} deleted successfully!")
def delpass():
    ovr = tk.Toplevel()
    ovr.geometry('500x500')
    v = StringVar()
    uweb = tk.Label(ovr, text="Website:")
    uweb.pack()
    upweb3 = tk.Entry(ovr, textvariable=v)
    upweb3.pack()
    def submit():
        website = v.get()
        delete_password(website)
    submit_button = tk.Button(ovr, text="Submit", command=submit)
    submit_button.pack()

def display_passwords():
    with open(fileloc, "r") as f:
        for line in f:
            print(line.strip())
```

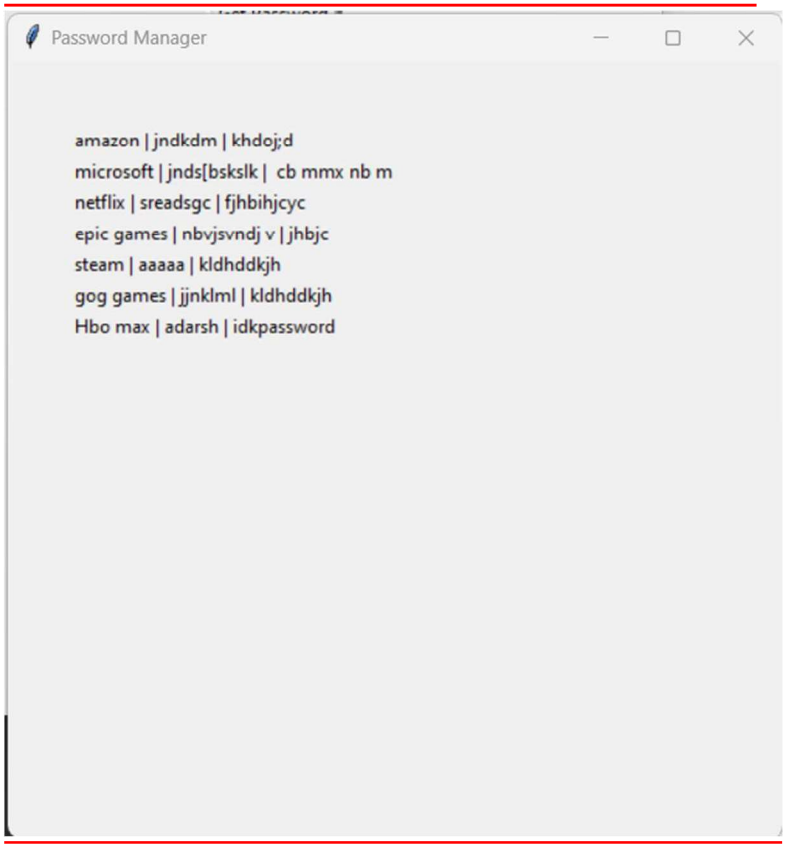
```
def dispass():
    j=tk. Toplevel()
    j.geometry('500x500')
    k=20
    with open(fileloc, "r") as f:
        for line in f:
            u=tk.Label(j,text=line).place(x=40,y=20+k)
            k+=20

win=tk.Tk()

win.title('Password Manager')
win.geometry('500x500')
btn = tk.Button(win, text = 'Add Password', bd = '5',
                        command = adpass)
btn.pack(side='top')
btn2 = tk.Button(win, text = 'Get Password', bd = '5',
                        command = getpass)
btn2.pack(side='top')
btn3= tk.Button(win, text = 'Delete Password', bd = '5',
                        command = delpass)
btn3.pack(side='top')
btn4 = tk.Button(win, text = 'Display Password', bd = '5',
                        command = dispass)
btn4.pack(side='top')
btn5 = tk.Button(win, text = 'Exit', bd = '5',
                        command = win.destroy)
btn5.pack(side='top')
win.mainloop()
```

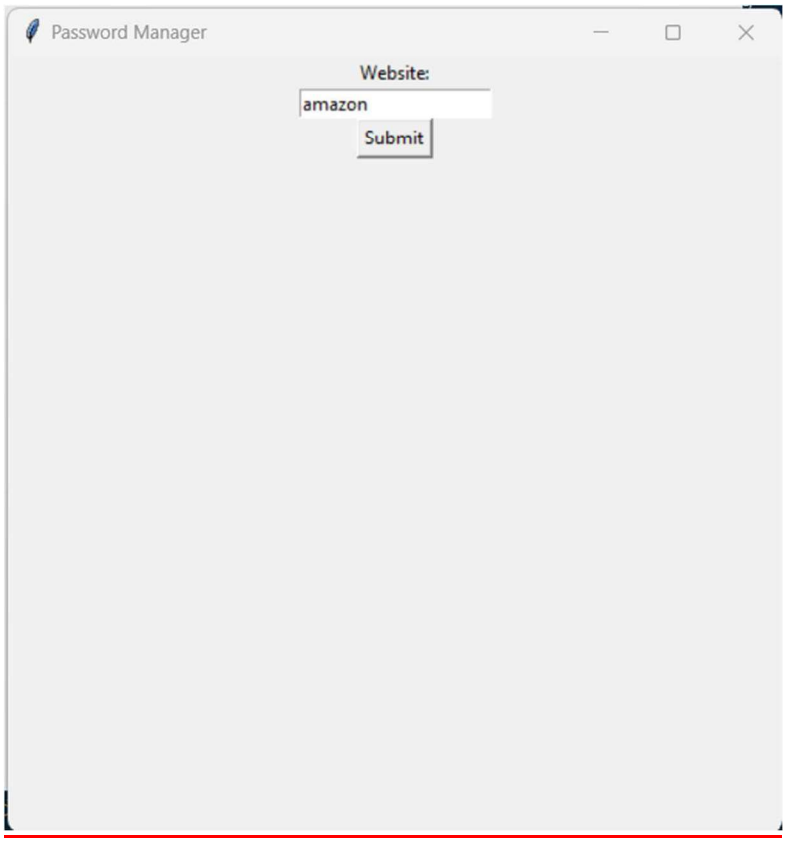
Output:


Password Manager
Add Password
Get Password
Delete Password
Display Password
Exit


Password Manager
Website:
Hbo max
Username:
adarsh
Password:
idkpassword
Submit


Password Manager
Website:
steam
Submit
kldhddkjh


Password Manager
amazon | jndkdm | khdoj;d
microsoft | jnds[bskslk | cb mmx nb m
netflix | sreadsgc | fjhbihjcyc
epic games | nbvjsvndj v | jhbjc
steam | aaaaa | kldhddkjh
gog games | jjnklml | kldhddkjh
Hbo max | adarsh | idkpassword



Code Report: Password Manager Application

Introduction

The provided code is a Python script that implements a basic Password Manager application using the tkinter library for the graphical user interface (GUI). This application allows users to perform various operations related to managing and storing passwords. Here's a breakdown of the code and its functionalities:

Code Structure

Import Statements

- The code begins by importing necessary modules: `os` and `tkinter`. It also imports everything (`*`) from the tkinter library.

File Location

- The `fileloc` variable is defined to store the file location for storing passwords. By default, it's set to `"D:\All coding\Python\passwords.txt"`. This location should be updated to a more appropriate directory.

Functions

1. `add_password(website, username, password)`: This function adds a new

password entry to the password file. It takes website, username, and password as arguments and appends them to the file.

2. `adpass()`: This function creates a GUI window for adding a password. It includes text input fields for the website, username, and password. When the "Submit" button is clicked, it calls the `add_password` function to add the password.

3. `get_password(website)`: This function retrieves the password associated with a given website from the password file.

4. `getpass()`: This function creates a GUI window for retrieving a password. It includes a text input field for the website. When the "Submit" button is clicked, it displays the password for the specified website.

5.`delete_password(website)`: This function deletes a password entry associated with a given website from the password file.

6. `delpass()`: This function creates a GUI window for deleting a password. It includes a text input field for the website. When the "Submit" button is clicked, it calls the `delete_password` function to delete the password.

7. `display_passwords()`: This function displays all passwords stored in the password file.

8. `dispass()`: This function creates a GUI window for displaying all passwords. It reads the passwords from the file and displays them in labels within the window.

Main Application
- The main application window (`win`) is created using tkinter. It includes buttons to perform the following actions:
  - Add a Password
  - Get a Password
  - Delete a Password
  - Display All Passwords
  - Exit the Application

Main Loop
- The `win.mainloop()` statement starts the main event loop, which listens for user interactions with the GUI and responds accordingly.

Functionality Overview

1. Add Password: Users can click the "Add Password" button to open a window where they can input a website, username, and password. Upon clicking "Submit," the password is added to the password file.

2. Get Password: Users can click the "Get Password" button to open a window where they can input a website. Upon clicking "Submit," the associated password is displayed.

3. Delete Password: Users can click the "Delete Password" button to open a window where they can input a website. Upon clicking "Submit," the password associated with the specified website is deleted from the file.

4. Display Passwords: Users can click the "Display Password" button to open a window that displays all passwords stored in the password file.

5. Exit: Users can click the "Exit" button to close the application.

Conclusion:

This software can be further enhanced. More features can be added for a better functionality