Design Computing Studio II Lab 2

Name: **Yuchen Shi** Student ID: **D23090120503**

Requirements:

1. Please submit the report in PDF format, and the file name is named "name_ID_HW2.pdf".
2. Please ensure the detergent code to intercept the map and the operation of the results of the cut-off map is clear and readable.
3. Please upload the file within the system regulation time, and attach the code file at the same time.

**Please submit assignments strictly in accordance with the above assignment requirements, otherwise you will be deducted points*

1、 Experiment topic: User login program design

2、 Objectives:

Through this lab, students will learn how to design and implement a simple user login program. Master basic input validation, security considerations, and implementation of program logic.

3、 Experiment:

1. Program Feature Requirements:

- The user enters the username and password to log in.
- Check whether the username and password entered are correct.
  - Username: name='root'
  - Password: passwd='admin'
- To prevent brute force attacks, limit the number of login attempts to three. If it is more than three times, the program should report an error message and terminate the login process.

2. **Code implementation:**

- Write complete code in a programming language you're familiar with, such as Python, C++, etc.
- Implement input validation and login logic.
- Use conditional statements and loop structures to control the number of logon attempts.

3. **Screenshot Requirements:**

- Once you have completed the procedure, simulate the login screen and test it.
- A screenshot of the submission program as it runs, including the interface for entering a username, password, and a successful or failed login.

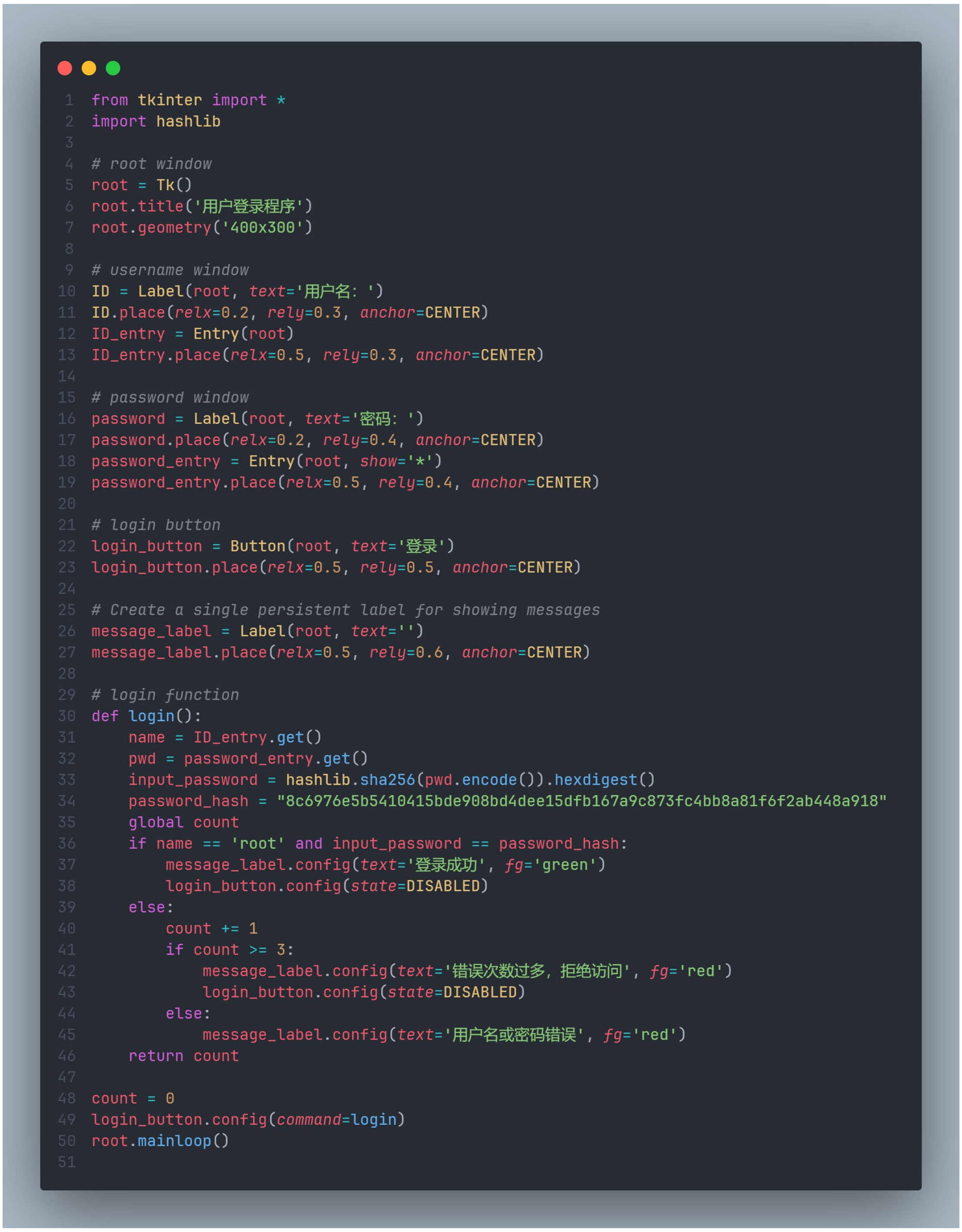
4. **Extra Tasks (Additional Questions):**

1) Consider and answer the following questions:
Why can't passwords be stored in plaintext? What are the security risks if we use plaintext storage? How to solve this problem? Please make a modification in your code to fix this.

2) Design an interface for your user login program.

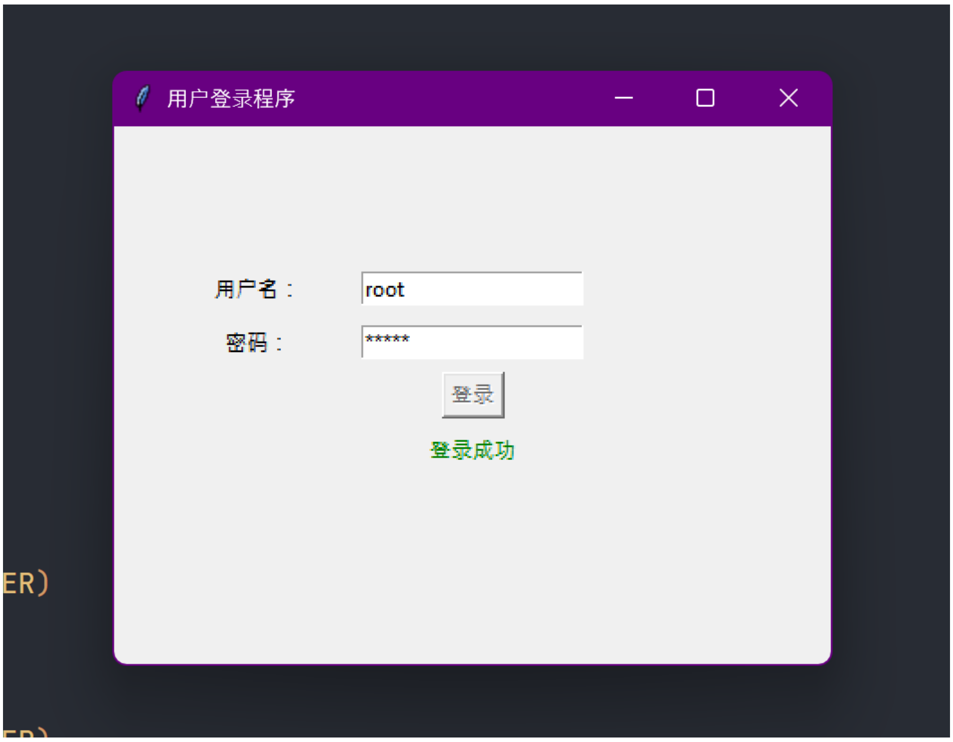
Answer sheets

- **Code Screenshot:**



```python
from tkinter import *
import hashlib

# root window
root = Tk()
root.title('用户登录程序')
root.geometry('400x300')

# username window
ID = Label(root, text='用户名：')
ID.place(relx=0.2, rely=0.3, anchor=CENTER)
ID_entry = Entry(root)
ID_entry.place(relx=0.5, rely=0.3, anchor=CENTER)

# password window
password = Label(root, text='密码：')
password.place(relx=0.2, rely=0.4, anchor=CENTER)
password_entry = Entry(root, show='*')
password_entry.place(relx=0.5, rely=0.4, anchor=CENTER)

# login button
login_button = Button(root, text='登录')
login_button.place(relx=0.5, rely=0.5, anchor=CENTER)

# Create a single persistent label for showing messages
message_label = Label(root, text='')
message_label.place(relx=0.5, rely=0.6, anchor=CENTER)

# login function
def login():
    name = ID_entry.get()
    pwd = password_entry.get()
    input_password = hashlib.sha256(pwd.encode()).hexdigest()
    password_hash = "8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918"
    global count
    if name == 'root' and input_password == password_hash:
        message_label.config(text='登录成功', fg='green')
        login_button.config(state=DISABLED)
    else:
        count += 1
        if count >= 3:
            message_label.config(text='错误次数过多，拒绝访问', fg='red')
            login_button.config(state=DISABLED)
        else:
            message_label.config(text='用户名或密码错误', fg='red')
    return count

count = 0
login_button.config(command=login)
root.mainloop()
```

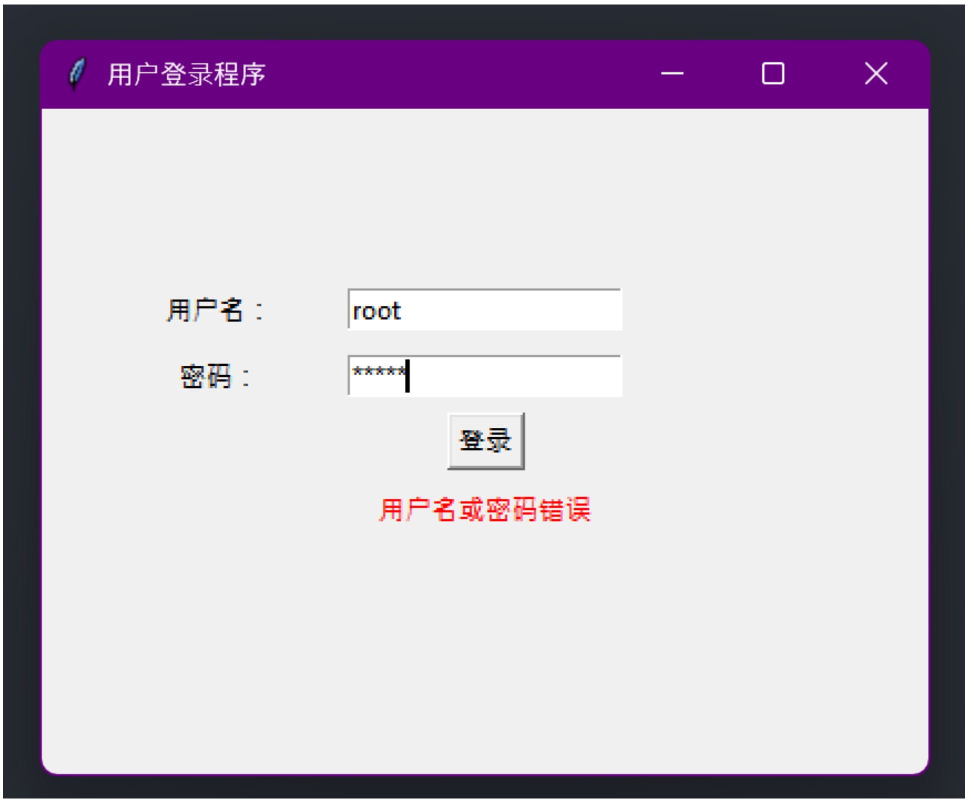
- **Screenshot of the running result:**



用户名： root

密码： *****

登录成功



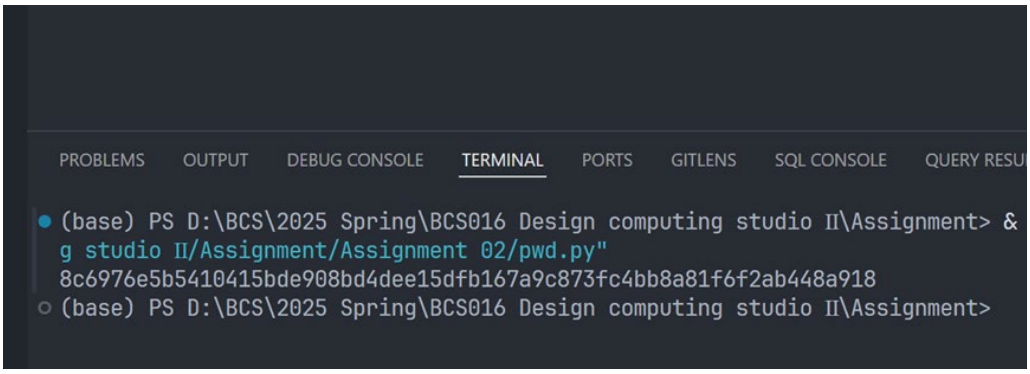
用户名： root

密码： *****

用户名或密码错误

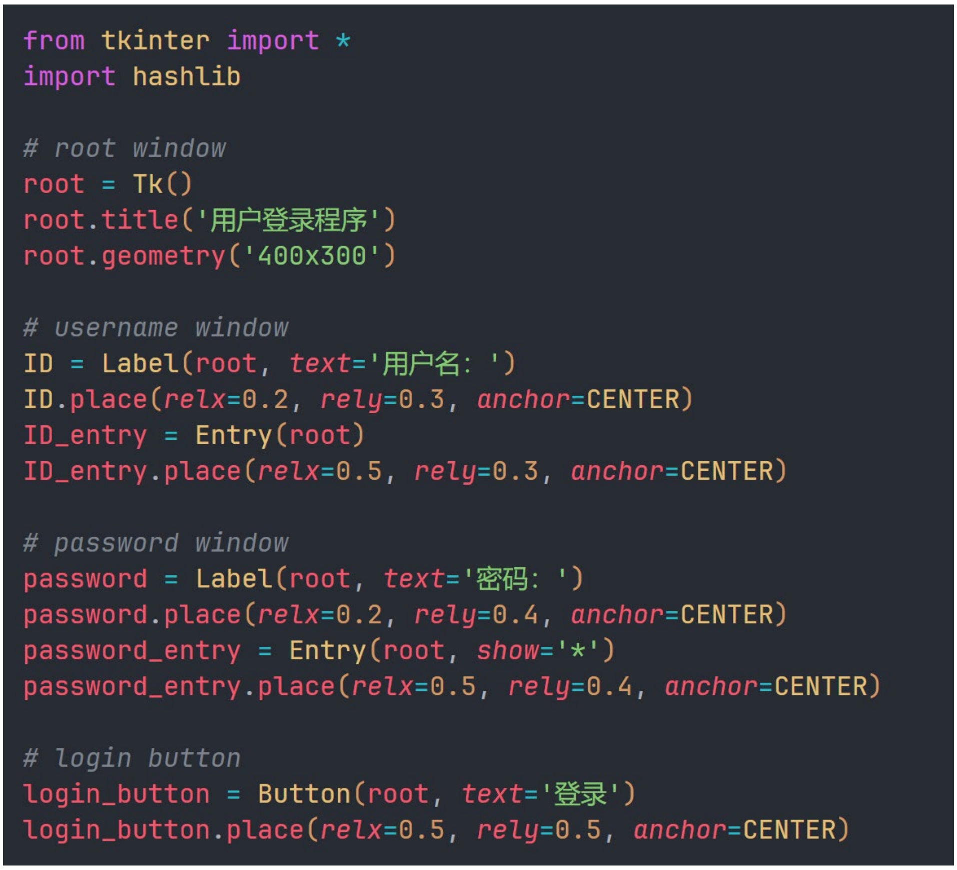
- **Screenshots of the answers to the additional questions and related codes:**

Use ciphertext to output the password - admin



```
● (base) PS D:\BCS\2025 Spring\BCS016 Design computing studio II\Assignment> &
  g studio II/Assignment/Assignment 02/pwd.py"
  8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
○ (base) PS D:\BCS\2025 Spring\BCS016 Design computing studio II\Assignment>
```

Login interface



```python
from tkinter import *
import hashlib

# root window
root = Tk()
root.title('用户登录程序')
root.geometry('400x300')

# username window
ID = Label(root, text='用户名: ')
ID.place(relx=0.2, rely=0.3, anchor=CENTER)
ID_entry = Entry(root)
ID_entry.place(relx=0.5, rely=0.3, anchor=CENTER)

# password window
password = Label(root, text='密码: ')
password.place(relx=0.2, rely=0.4, anchor=CENTER)
password_entry = Entry(root, show='*')
password_entry.place(relx=0.5, rely=0.4, anchor=CENTER)

# login button
login_button = Button(root, text='登录')
login_button.place(relx=0.5, rely=0.5, anchor=CENTER)
```

(1) If password were stored in plaintext, attackers can easily read and use the password when the database is attacked. Additionally, individuals with database access can retrieve the password through the backend, leading to privacy breaches. In short, storing password in plaintext poses significant risks, as it greatly increases the chances of personal information being exposed or attacked.

In that situation, password could be stored using secure hashing algorithms. Use a cryptographic hash function like bcrypt, Argon2, or PBKDF2 to hash the passwords, another method is salting, which adds a unique salt to each password before hashing to prevent rainbow table attacks.

(2) I used tkinter package in Python to design a login interface that includes input fields and basic prompt messages.

Jason