

# readme

倪浚桐

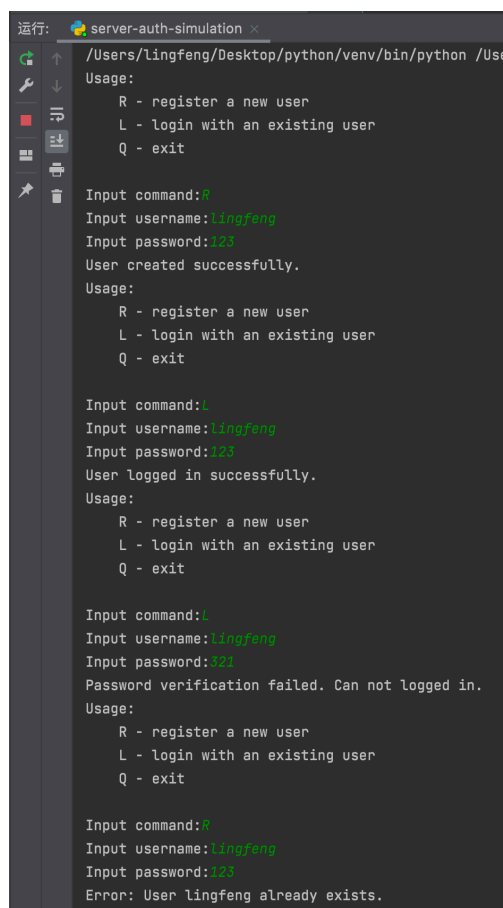
202022161224

Lab7-progarm3

macOS Monterey 12.0.1

Pycharm 11.0.12 x86-64

Python 3.9.5



```
运行: server-auth-simulation x
/Users/lingfeng/Desktop/python/venv/bin/python /Usa
Usage:
  R - register a new user
  L - login with an existing user
  Q - exit

Input command:
Input username:lingfeng
Input password:123
User created successfully.
Usage:
  R - register a new user
  L - login with an existing user
  Q - exit

Input command:
Input username:lingfeng
Input password:123
User logged in successfully.
Usage:
  R - register a new user
  L - login with an existing user
  Q - exit

Input command:
Input username:lingfeng
Input password:123
Password verification failed. Can not logged in.
Usage:
  R - register a new user
  L - login with an existing user
  Q - exit

Input command:
Input username:lingfeng
Input password:123
Error: User lingfeng already exists.
```

```
1 from __future__ import annotations
2 from typing import Dict
3 import os
4 import getpass
5 import hashlib
6
7 database: Dict[str, UserPassword] = dict()
8
9
```

```

10 class UserPassword:
11     def __init__(self):
12         self.username: str = ''
13         self.password_hash: bytes = b''
14         self.salt: bytes = b''
15         self.method: str = 'scrypt'
16
17     def verify_password(self, password: str) -> bool:
18         # TODO
19         password_bytes: bytes = bytes(password, encoding="utf8")
20         password_hash: bytes = hashlib.scrypt(password_bytes, salt=self.salt, n=4,
r=8, p=16)
21         if password_hash == self.password_hash:
22             return True
23
24
25 def database_add_item(user: UserPassword) -> None:
26     if user.username in database:
27         raise Exception('User {} already exists.'.format(user.username))
28     database[user.username] = user
29
30
31 def login_user(username: str, password_plaintext: str) -> bool:
32     if username not in database:
33         raise Exception('User {} does not exist.'.format(username))
34     return database[username].verify_password(password_plaintext)
35
36
37 def register_user(username: str, password_plaintext: str) -> None:
38     # TODO
39     password_bytes: bytes = bytes(password_plaintext, encoding="utf8")
40     salt_bytes: bytes = os.urandom(21)
41     password_hash: bytes = hashlib.scrypt(password_bytes, salt=salt_bytes, n=4,
r=8, p=16)
42     user = UserPassword()
43     user.username = username
44     user.password_hash = password_hash
45     user.salt = salt_bytes
46     database_add_item(user)
47
48
49 if __name__ == '__main__':
50     while True:
51         try:
52             print('Usage:')
53             print('\tR - register a new user')
54             print('\tL - login with an existing user')
55             print('\tQ - exit')
56             print('')

```

```
57         command: str = input('Input command:')
58         if command == 'Q':
59             exit(0)
60         elif command == 'R' or command == 'L':
61             username: str = input('Input username:')
62             # password: str = getpass.getpass('Input password:') # will not
work properly for PyCharm, IDLE, etc.
63             password: str = input('Input password:')
64             if command == 'R':
65                 register_user(username, password)
66                 print('User created successfully.')
67             elif command == 'L':
68                 login_valid: bool = login_user(username, password)
69                 if login_valid:
70                     print('User logged in successfully.')
71                 else:
72                     print('Password verification failed. Can not logged in.')
73             else:
74                 assert False
75         else:
76             raise Exception('Invalid command.')
77     except Exception as e:
78         print('Error: {}'.format(e))
```