**IMPLEMENTATION – EX 8 :**

from getpass import getpass

import re

import pickle

import sys

from datetime import datetime

import json

class User:

"""Represents a user with a username and password."""

def \_\_init\_\_(self, username: str, password: str):

self.username = username

self.password = password

def check\_password\_strength(password):

"""Check if the password meets certain strength criteria."""

password\_pattern = "^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9])(?=.\*?[#?!@$%^&\*-]).{8,}$"

if re.match(password\_pattern, password):

return True

return False

def serialize\_user(user: User):

"""Serialize and store a user object in a file."""

with open("users.pickle", "ab") as users\_file:

pickle.dump(user, users\_file)

def check\_credentials(username, password):

"""Check if the provided username and password match a registered user."""

with open("users.pickle", "rb") as users\_reader:

while True:

try:

user = pickle.load(users\_reader)

except EOFError:

return False

if user.username == username and user.password == password:

return True

else:

continue

def deserialize\_users():

"""Deserialize and yield user objects from a file."""

with open("users.pickle", "rb") as users\_reader:

while True:

try:

user = pickle.load(users\_reader)

except EOFError:

return False

yield user.username

def jsonify(message\_str):

"""Serialize and store chat messages in a JSON file."""

try:

with open("messages.json", "r") as file:

chat\_data = json.load(file)

except FileNotFoundError:

chat\_data = {}

for username, messages in message\_str.items():

if username in chat\_data:

chat\_data[username].extend(messages)

else:

chat\_data[username] = messages

with open("messages.json", "w") as message\_writer:

json.dump(chat\_data, message\_writer, indent=4)

def display\_messages(chat\_data):

"""Display chat messages for the current user."""

username = ChatApp.current\_user

if username in chat\_data:

messages = chat\_data[username]

print(f"Messages for {username}")

for message in messages:

timestamp = message["timestamp"]

message\_text = message["message"]

status = "Sent:\t\t" if message["type"] == "sent" else "Received:\t"

if message.get("to"):

display\_name = f"To {message.get('to')}"

else:

display\_name = f"From {message.get('from')}"

print(f"{status}{timestamp} - {display\_name}: {message\_text}")

else:

print(f"No messages found for {username}")

def search\_messages(chat\_data, content):

"""Search for messages containing a specific phrase."""

matching\_messages = []

messages = chat\_data[ChatApp.current\_user]

for message in messages:

if content in message["message"]:

matching\_messages.append(message)

return matching\_messages

class ChatApp:

"""A simple chat application."""

usernames = []

current\_user = None

def \_\_init\_\_(self):

"""Initialize the chat application."""

user\_db = open("users.pickle", "ab")

user\_db.close()

ChatApp.usernames = list(deserialize\_users())

def add\_user(self):

"""Create a new user and store their credentials."""

username = input("Please enter a username: ")

print(

"Note that your password contains minimum 8 characters and contain atleast\n"

"1 uppercase character, 1 lowercase character,\n"

"1 digit and 1 special character."

)

password = getpass("Please enter a password: ")

re\_password = getpass("Please re-enter your password: ")

if password != re\_password:

print("Passwords are not the same!")

return False

elif username in ChatApp.usernames:

print("Username already exists!")

return False

else:

if not check\_password\_strength(password):

print("Password is not strong enough!")

return False

else:

user = User(username, password)

ChatApp.usernames.append(user.username)

serialize\_user(user)

return True

def login(self):

"""Log in with a username and password."""

if ChatApp.current\_user is not None:

print("User already logged in! Please log out before logging in!")

return False

username = input("Please enter the username: ")

password = getpass("Please enter the password: ")

if username not in ChatApp.usernames:

print("Invalid credentials! User not registered.")

return False

else:

is\_valid = check\_credentials(username, password)

if not is\_valid:

print("Invalid credentials!")

return False

else:

ChatApp.current\_user = username

return True

def send\_message(self, user\_choice, message\_content):

"""Send a message to another user."""

time\_stamp = datetime.now().strftime("%d/%m/%Y, %H:%M:%S")

sender = ChatApp.current\_user

receiver = ChatApp.usernames[user\_choice]

message = {}

message[sender] = []

message[receiver] = []

contents\_send = {

"message": message\_content,

"timestamp": time\_stamp,

"to": receiver,

"type": "sent",

}

contents\_rec = {

"message": message\_content,

"timestamp": time\_stamp,

"from": sender,

"type": "received",

}

message[sender].append(contents\_send)

message[receiver].append(contents\_rec)

jsonify(message)

def display\_all\_messages(self):

"""Display all chat messages for the current user."""

try:

with open("messages.json", "r") as file:

chat\_data = json.load(file)

except FileNotFoundError:

chat\_data = {}

display\_messages(chat\_data)

def search\_message(self):

"""Search for messages containing a specific phrase and display the results."""

if chatapp.current\_user is None:

print("You have not logged in! Log in first.")

return

search\_phrase = input("Enter the message phrase to search: ")

try:

with open("messages.json", "r") as file:

chat\_data = json.load(file)

except FileNotFoundError:

chat\_data = {}

messages = search\_messages(chat\_data, content=search\_phrase)

if messages:

for message in messages:

timestamp = message["timestamp"]

message\_text = message["message"]

status = "Sent:\t\t" if message["type"] == "sent" else "Received:\t"

if message.get("to"):

display\_name = f"To {message.get('to')}"

else:

display\_name = f"From {message.get('from')}"

print(f"{status}{timestamp} - {display\_name}: {message\_text}")

else:

print("No messages found!")

def log\_out(self):

"""Log out the current user."""

ChatApp.current\_user = None

def exit(self):

"""Exit the chat application."""

ChatApp.current\_user = None

sys.exit(0)

if \_\_name\_\_ == "\_\_main\_\_":

chatapp = ChatApp()

while True:

print("=" \* 26)

print("1. Create a user")

print("2. Login")

print("3. Chat")

print("4. Display messages")

print("5. Log out")

print("6. Search for a message")

print("7. Exit")

ch = input("Enter your choice:").strip()

if ch == "1":

chk = chatapp.add\_user()

if chk:

print("User created successfully!")

else:

continue

elif ch == "2":

chk = chatapp.login()

if chk:

print("User logged in successfully!")

print(f"Current User: {chatapp.current\_user}")

else:

continue

elif ch == "3":

if chatapp.current\_user is None:

print("You have not logged in! Log in first.")

continue

print("Choose the user you want to chat with:")

for idx, user in enumerate(ChatApp.usernames):

print(idx + 1, "\t", user)

chat\_choice = int(input("Enter the choice: "))

if not ((chat\_choice >= 1) and (chat\_choice <= len(ChatApp.usernames))):

print("Please enter the correct choice!")

continue

message\_content = input("Please provide the message to be sent: ")

chatapp.send\_message(chat\_choice - 1, message\_content)

elif ch == "4":

if chatapp.current\_user is None:

print("You have not logged in! Log in first.")

continue

chatapp.display\_all\_messages()

elif ch == "5":

chatapp.log\_out()

print("Logged out successfully.")

elif ch == "6":

if chatapp.current\_user is None:

print("You have not logged in! Log in first.")

continue

chatapp.search\_message()

elif ch == "7":

print("Exiting...")

chatapp.exit()

**OUTPUT:**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:1**

**Please enter a username: ram**

**Note that your password contains minimum 8 characters and contain atleast**

**1 uppercase character, 1 lowercase character,**

**1 digit and 1 special character.**

**Please enter a password:**

**Please re-enter your password:**

**User created successfully!**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:1**

**Please enter a username: anbu**

**Note that your password contains minimum 8 characters and contain atleast**

**1 uppercase character, 1 lowercase character,**

**1 digit and 1 special character.**

**Please enter a password:**

**Please re-enter your password:**

**User created successfully!**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:2**

**Please enter the username: ram**

**Please enter the password:**

**User logged in successfully!**

**Current User: ram**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:3**

**Choose the user you want to chat with:**

**1 ram**

**2 anbu**

**Enter the choice: 1**

**Please provide the message to be sent: Hello Ram, How are you...**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:4**

**Messages for ram**

**Sent: 17/11/2023, 15:35:31 - To ram: Hello Ram, How are you...**

**Received: 17/11/2023, 15:35:31 - From ram: Hello Ram, How are you...**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:6**

**Enter the message phrase to search: Ram**

**Sent: 17/11/2023, 15:35:31 - To ram: Hello Ram, How are you...**

**Received: 17/11/2023, 15:35:31 - From ram: Hello Ram, How are you...**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:5**

**Logged out successfully.**

**==========================**

**1. Create a user**

**2. Login**

**3. Chat**

**4. Display messages**

**5. Log out**

**6. Search for a message**

**7. Exit**

**Enter your choice:7**

**Exiting...**