### LIBRARIES ###

try:

import hashlib as h

import datetime as d

import secrets as s

import os

except ImportError as e:

print(f"Error: {e}. Please make sure the required libraries are installed.")

exit(1)

### END LIBRARIES ###

### FUNCTIONS ###

def randomInt(bits): # Returns a strong random integer of "bits" bits

try:

num = s.randbits(bits)

return num

except Exception as e:

print(f"Error: {e}. Please try again.")

def randomStr(x): # Returns a strong random string of length "x" of characters selected from "dictionary"

try:

dictionary = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890-\_+=@[](),./#~!£$%^&\*:;"

return ''.join(s.choice(dictionary) for \_ in range(x))

except Exception as e:

print(f"Error: {e}. Please try again.")

def gen\_sha3512\_hash(x): # Returns the SHA3-512 hash of "inp"

try:

obj = h.sha3\_512()

obj.update(x.encode())

return str(obj.hexdigest())

except Exception as e:

print(f"Error: {e}. Please try again.")

def genSalt(): # Returns a strongly random 512-bit salt

try:

randomness = str(randomInt(4096)) + randomStr(4096)

salt = gen\_sha3512\_hash(randomness)

return salt

except Exception as e:

print(f"Error: {e}. Please try again.")

def addUser(username, password): # Hashes, salts, and stores credentials in their respective files

try:

usernames = []

if not os.path.isfile("pass.db"):

open("pass.db", "w").close()

if not os.path.isfile("salt.db"):

open("salt.db", "w").close()

if not os.path.isfile("user.db"):

open("user.db", "w").close()

with open("user.db", "rt") as userFile: # Fills the array "usernames"

for userLine in userFile:

usernames.append(userLine.strip("\n"))

with open("pass.db", "a") as p, open("salt.db", "a") as s, open("user.db", "a") as u:

salt = genSalt() # Generate hashes to store

userHash = gen\_sha3512\_hash(username)

passSalt = password + salt

passHash = gen\_sha3512\_hash(passSalt)

available = True

for i in range(len(usernames)):

if usernames[i] == userHash: # Check if the username is available

available = False

if available:

p.write(passHash + "\n") # Write values to files

s.write(salt + "\n")

u.write(userHash + "\n")

print("Your credentials have been added!\n")

else:

print("That username is taken! Please try again...\n")

except Exception as e:

print(f"Error: {e}. Please try again.")

def login(username, password): # Verifies the user entered "username" and "password," and prints "Logged in!" if they're in the database

# Test Credentials: "testUser", "testPass"

try:

usernames = [] # Define credential lists

passwords = []

salts = []

if not os.path.isfile("pass.db"):

open("pass.db", "w").close()

if not os.path.isfile("salt.db"):

open("salt.db", "w").close()

if not os.path.isfile("user.db"):

open("user.db", "w").close()

with open("pass.db", "rt") as passFile: # Fill lists with credentials

for passLine in passFile:

passwords.append(passLine.strip("\n"))

with open("salt.db", "rt") as saltFile:

for saltLine in saltFile:

salts.append(saltLine.strip("\n"))

with open("user.db", "rt") as userFile:

for userLine in userFile:

usernames.append(userLine.strip("\n"))

# Generate "username" hash for comparison & lookup

userHash = gen\_sha3512\_hash(username)

salt = ""

passHash = ""

location = 0 # Location of credentials in the list

found = False

for i in range(len(usernames)): # Look up location of credentials

if usernames[i] == userHash:

location = i

found = True

break

if found: # Look up the rest of the credentials

salt = salts[location]

passSalt = password + salt

passHash = gen\_sha3512\_hash(passSalt)

# Verify the credentials and return True if correct

if userHash == usernames[location] and passHash == passwords[location]:

print("Logged in!\n")

else:

print("Credentials not found! Please try again...\n")

except Exception as e:

print(f"Error: {e}. Please try again.")

### END FUNCTIONS ###

### MAIN PROGRAM ###

go = True

while go:

try:

option = input("Enter 1 to sign up\nEnter 2 to log in\nEnter 3 to exit\n: ")

if option.isdigit():

option = int(option)

else:

print("Invalid input. Please enter a number.")

continue

if option == 1:

username = input("\nPlease enter a username: ")

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

addUser(username, password)

elif option == 2:

username = input("\nPlease enter your username: ")

password = input("Please enter your password: ")

login(username, password)

elif option == 3:

go = False

else:

print("Invalid option. Please enter a valid option.")

except Exception as e:

print(f"Error: {e}. Please try again.\n")

### END MAIN PROGRAM ###