Oluseyi Kataye

Prof. Jackson

CTEC 445

4/30/25

Assignment: Secure Password Generation and Multi-Factor Authentication (MFA)

Source code:

# Oluseyi Kataye #

# Prof. Jackson #

# CTEC 445 #

# 5/1/2025 #

# Assignment: Secure Password Generation and Multi-Factor Authentication (MFA) #

import random, datetime, time

#Function 1: Password Generator

def password\_generation(passw):

print ("Your Secure Password is: "+''.join(random.choice('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890~`!@#$%^&\*()\_-+={[}]|:;"\'<,>.?/') for \_ in range(12)))

#Function 2: Password Verification

def password\_verification():

errors = []

specchars = r"""~`!@#$%^&\*()\_-+={[}]|\:;"'<,>.?/"""

passw = input("Please create a password: ")

if(len(passw)<8):

print("\nWeak password. Please use more characters.")

elif(len(passw)<=12):

print("\nModerate password. Good; a bit more characters will make this a stronger password.")

elif(len(passw)>=14):

print("\nStrong password.")

if len(passw) < 12:

errors.append("\nNot enough chars (12 minimum).")

if not any(char.isupper() for char in passw):

errors.append("This password needs at least one uppercase letter.")

if not any(char.islower() for char in passw):

errors.append("This password needs at least one lowercase letter.")

if not any(char.isdigit() for char in passw):

errors.append("This password needs at least one number.")

if not any(char in specchars for char in passw):

errors.append("This password needs at least one special character (such as:"+specchars+").")

for errorms in errors:

print(errorms)

if not errors:

print("Great! This is a Strong and Great password.")

return errors

#Function 3: Multi-Factor Authentication with One-Time-Password

def multifactorauthen\_otp(passw):

passw = ''.join(random.choice('1234567890') for \_ in range(6))

print("Your One-Time-Password is: " + passw)

print("Your One-Time-Password will expire in 60 secs")

starttime = datetime.datetime.now()

expireotp = starttime + datetime.timedelta(seconds=60)

verifyp = input("\nVerify the OTP: ")

otp = datetime.datetime.now()

if otp > expireotp:

print("Your OTP has expired.")

elif verifyp == passw:

print("Your OTP has been verified.")

else:

print("Your OTP could not be verified.")

spassw = input("Press any key to generate a Secure Password: ")

password\_generation(spassw)

passwv = input("\nPress any key to start the password verifier where you'll input a password, and it's strength will be tested: ")

password\_verification()

otpassw = input("\nPress any key to generate a 6 digit pin: ")

multifactorauthen\_otp(otpassw)