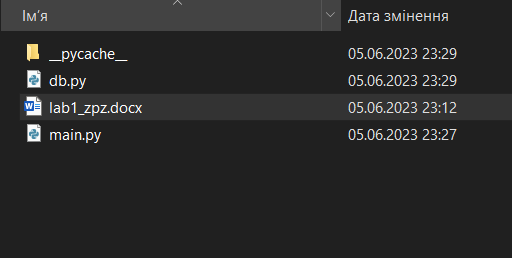
Виконала Музичка-Скрипка Олександра

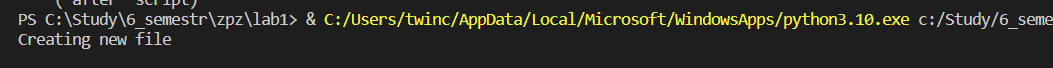
Варіант 10

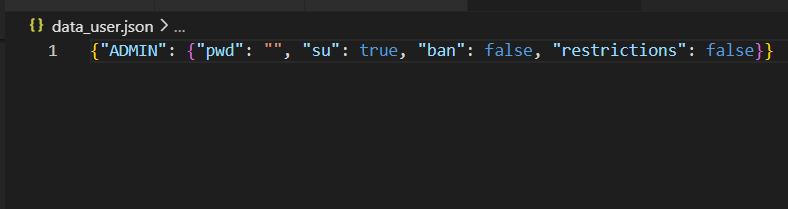
Наявність рядкових і прописних букв, а також знаків арифметичних операцій.

Дані про юзерів зберігаються в файлі data\_user.json

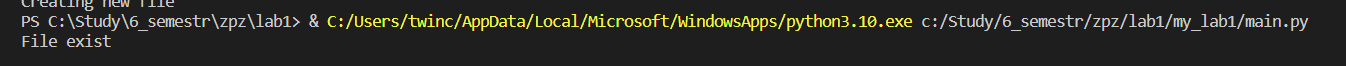
Файл створився при першому запуску коду



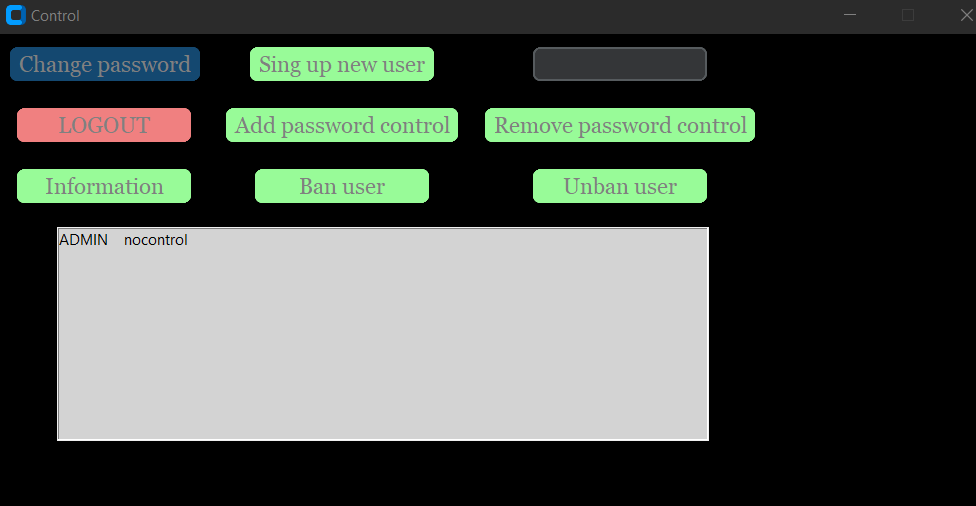




При повторному підключенні

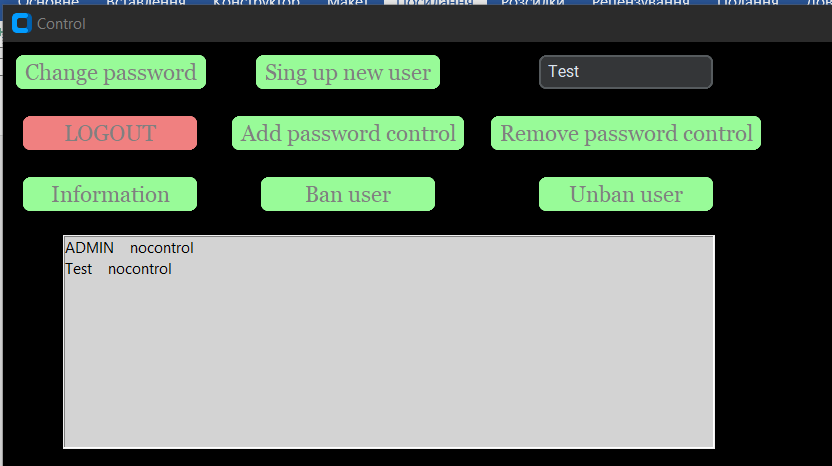


Режим Адміністратора

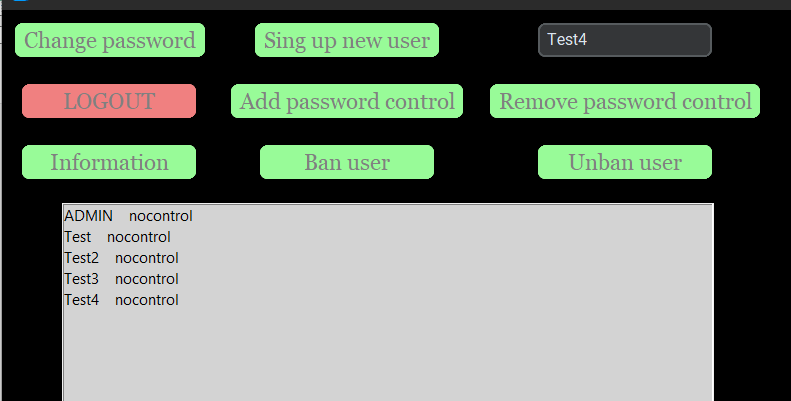


Додамо користувача Test

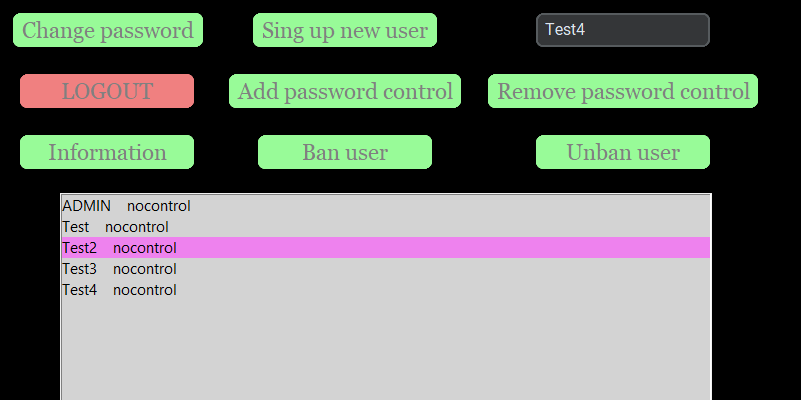




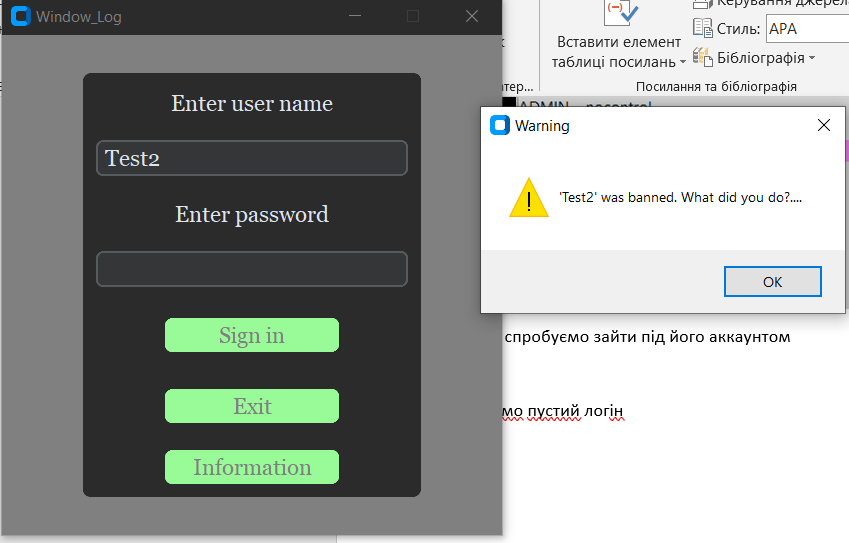
І ще декілька



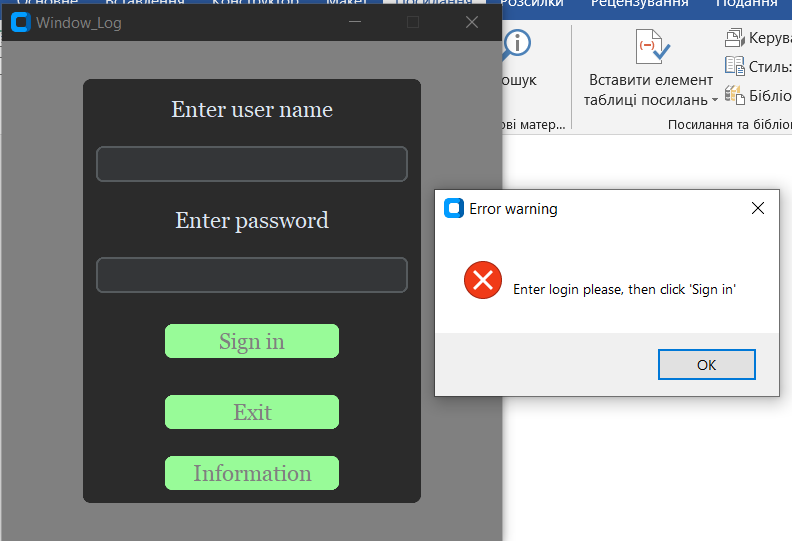
Заблокуємо Test2



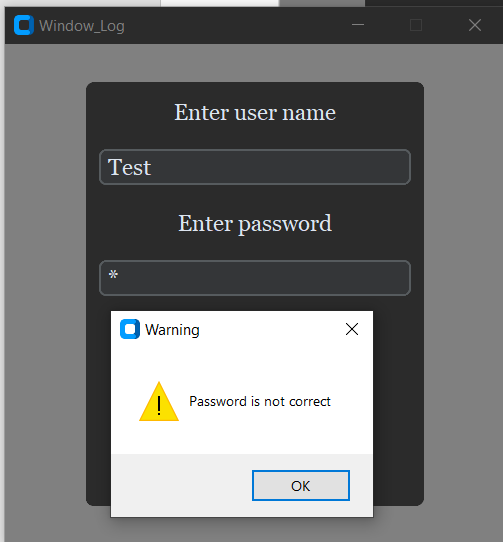
Тепер спробуємо зайти під його аккаунтом



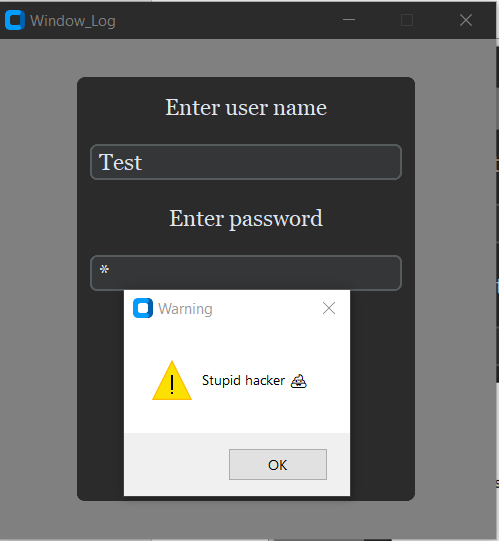
Введемо пустий логін



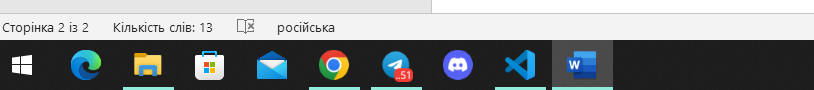
Неправильний пароль



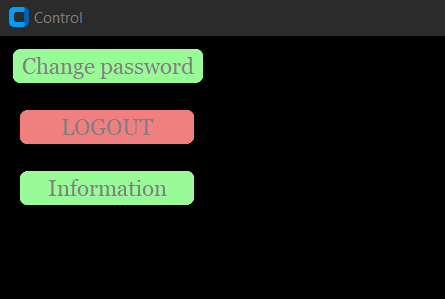
Якщо тричі неправильно ввести пароль



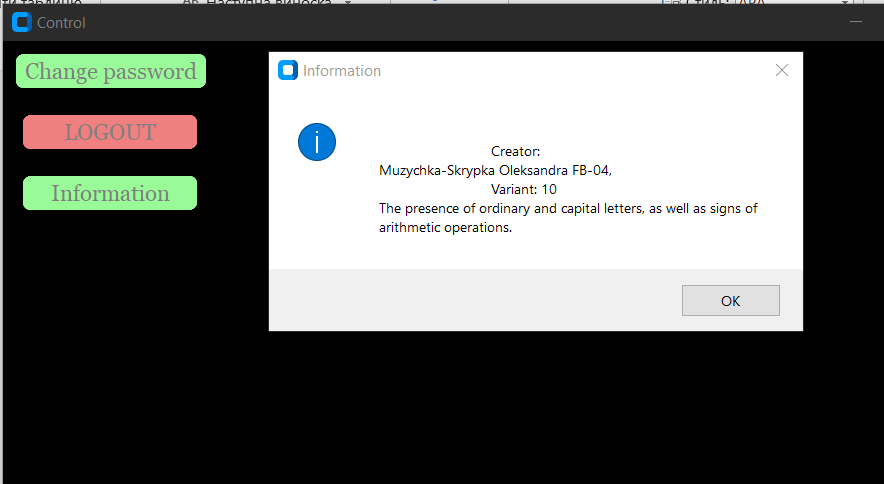
Потім программа закрилась



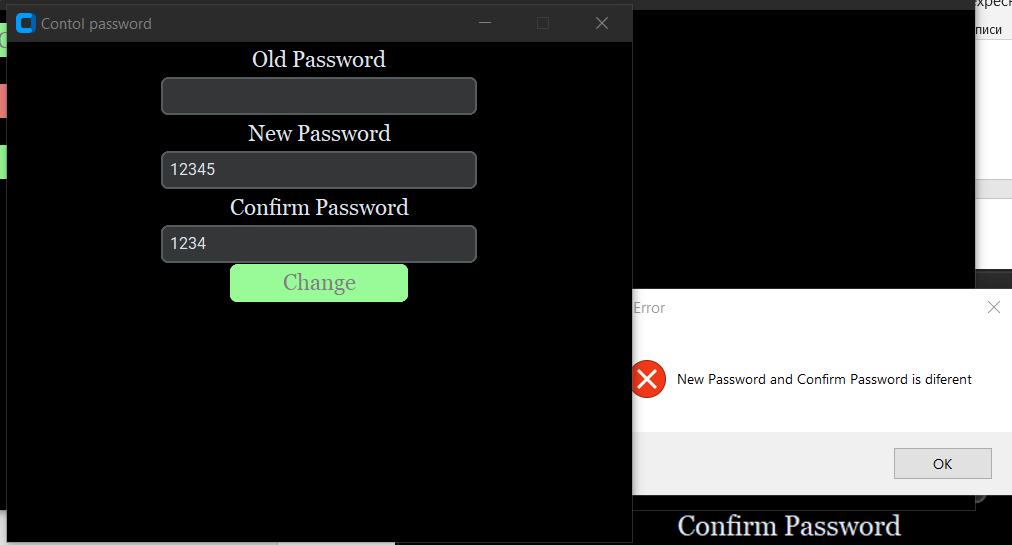
Режим користувача

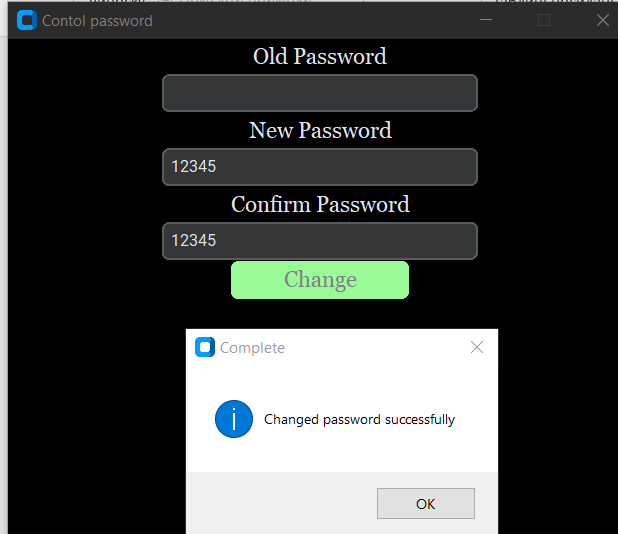


Інформація

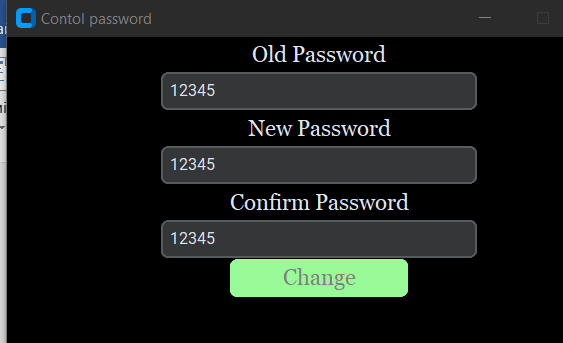


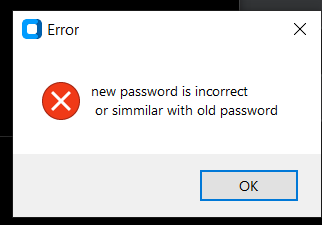
Спробуємо змінити пароль



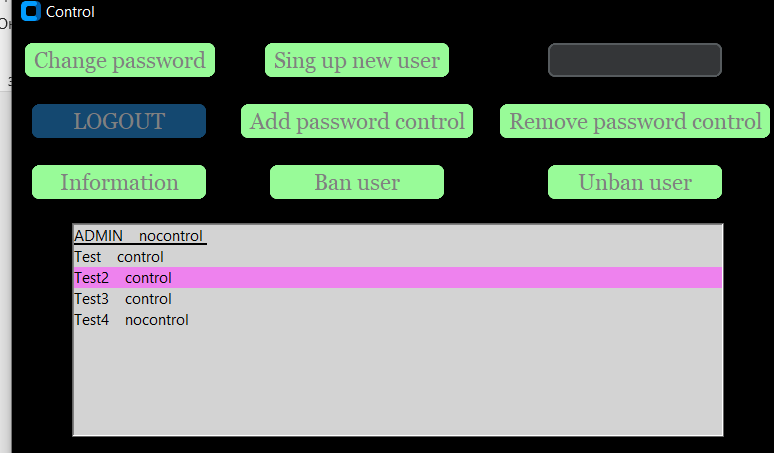


Тепер спробуємо замінити на такий самий пароль

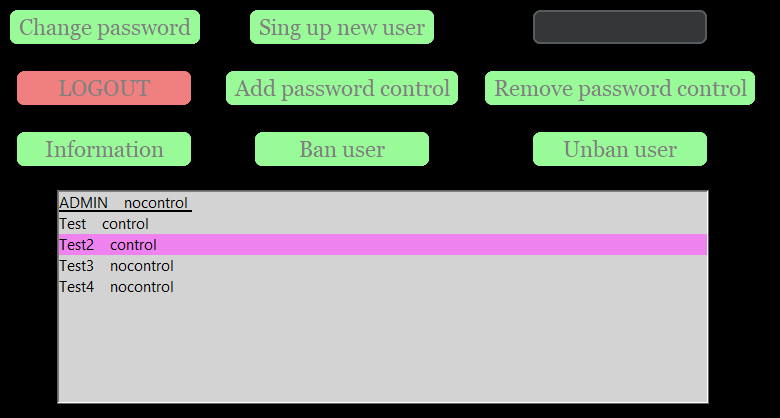




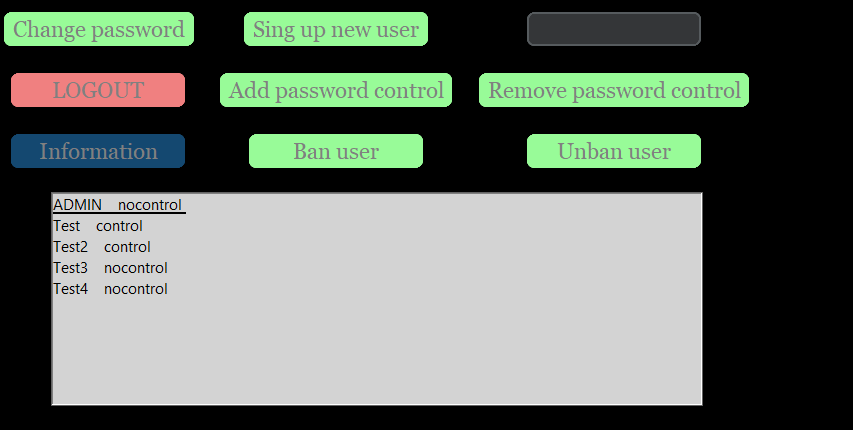
Повернемось режиму адміна та встановимо обмеження щодо паролю для юзера Test та Test3



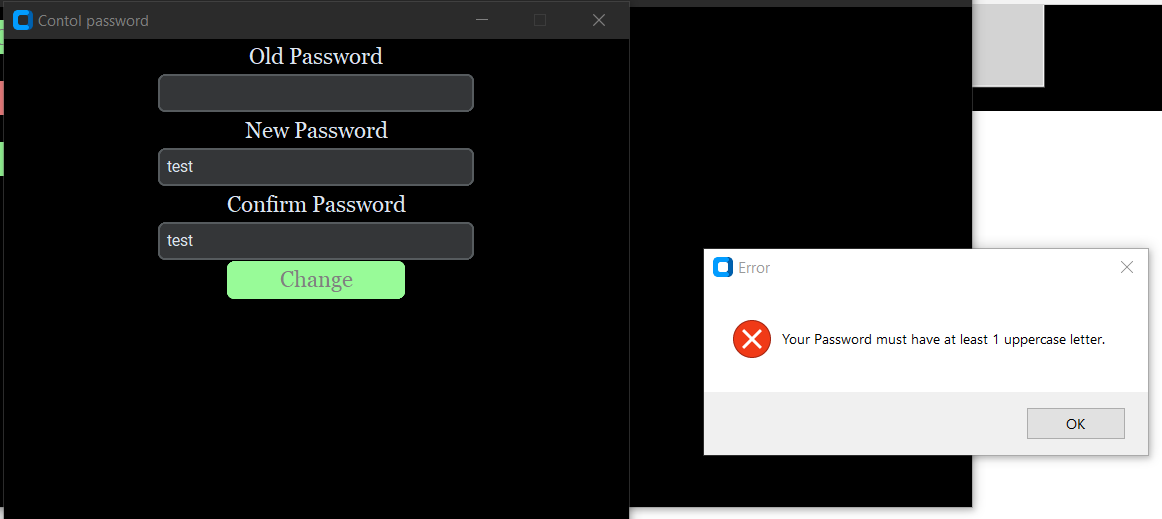
Видалимо обмеження для Test3

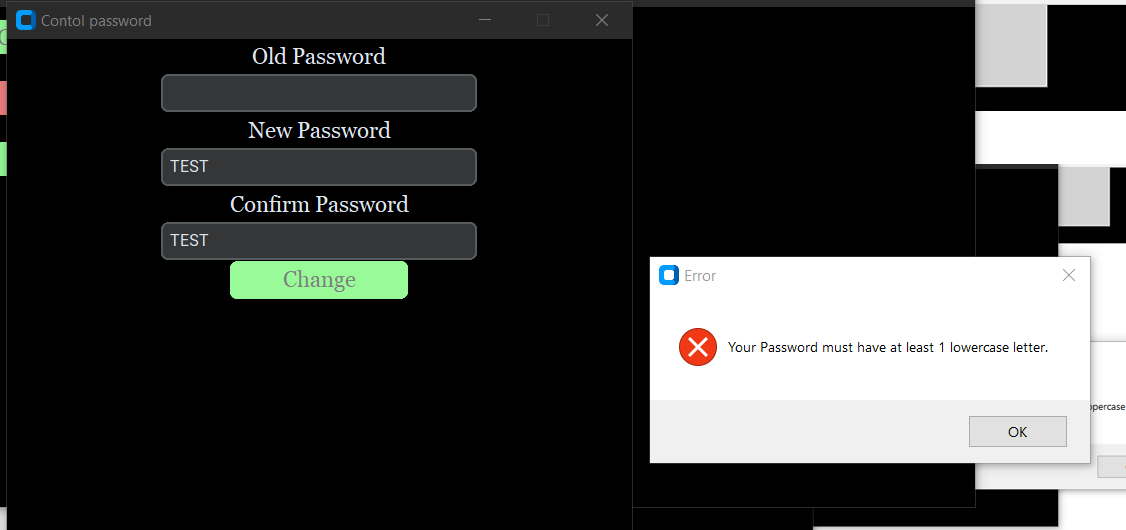


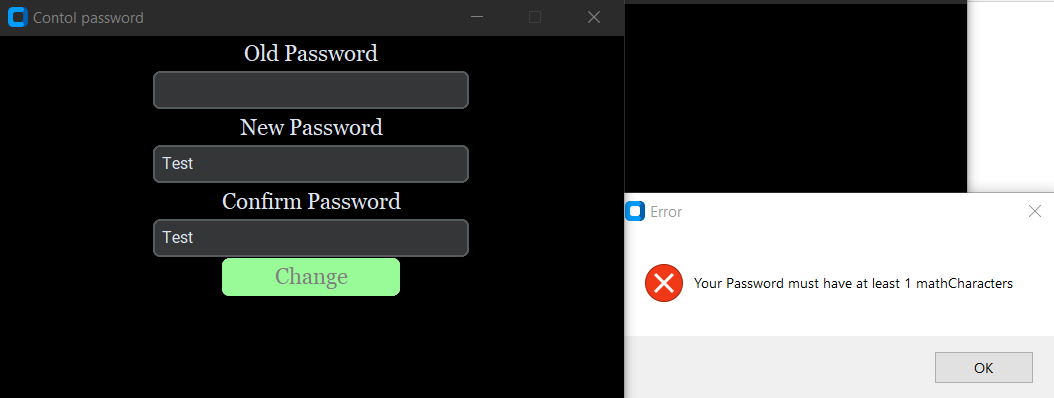
Розбанимо Test2

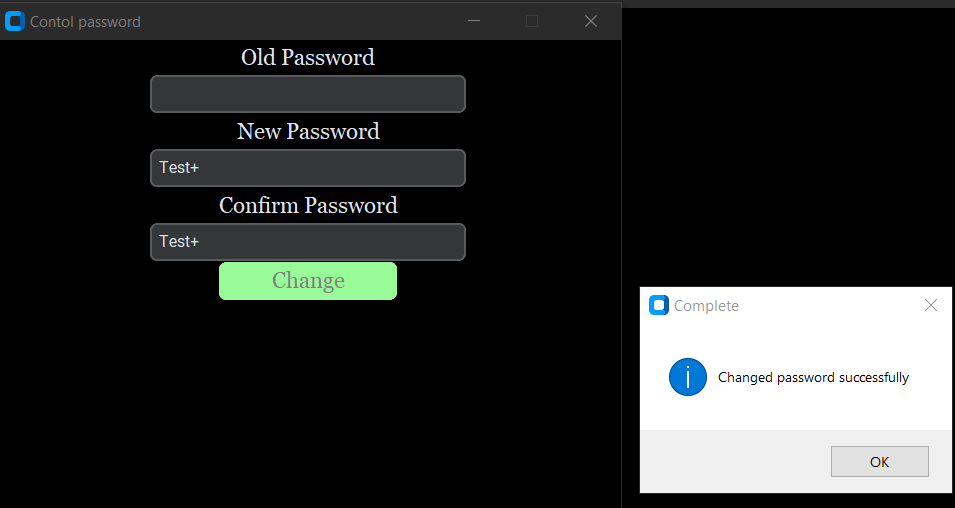


Перейдемо знову до користувача Test

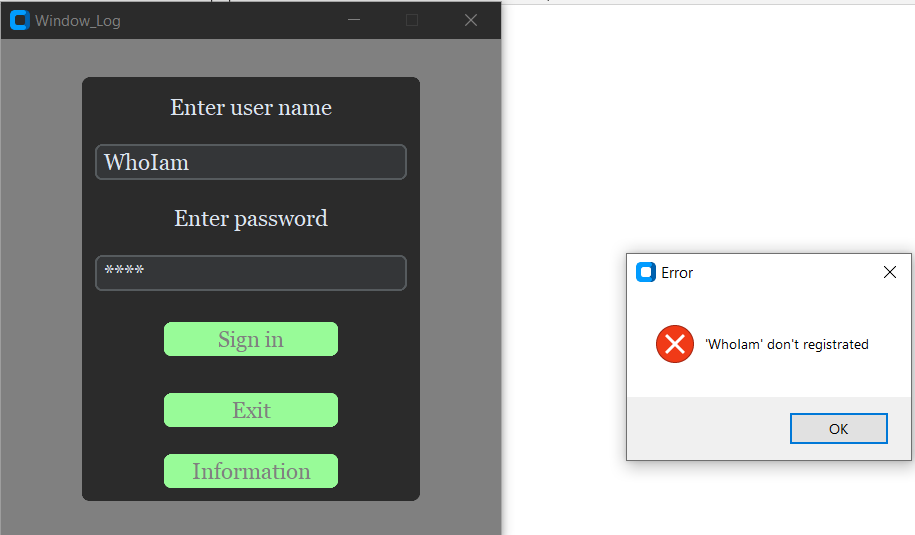




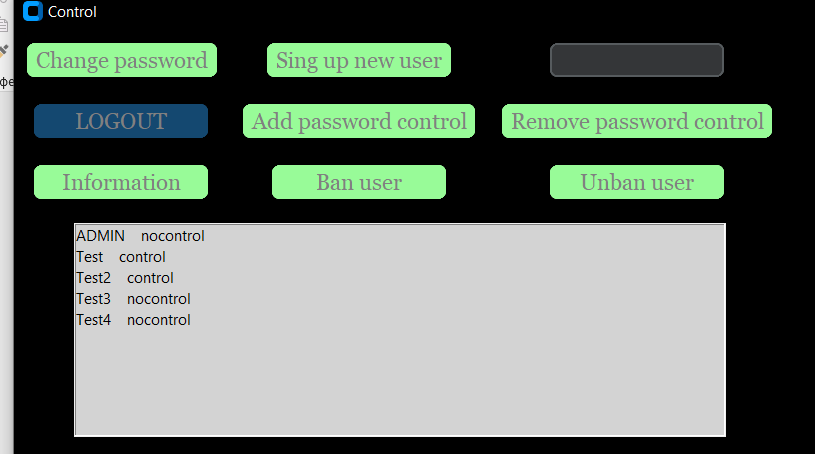




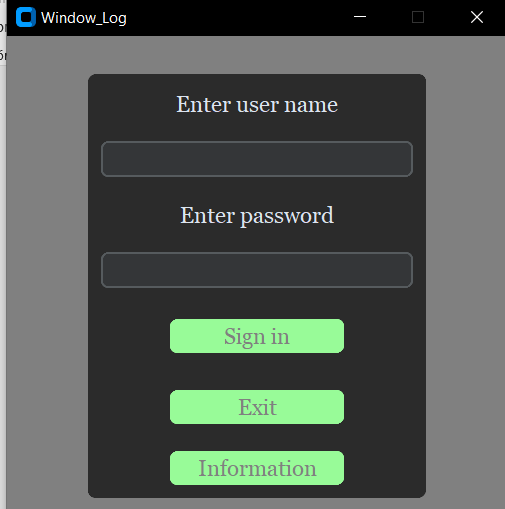
Спробуємо зайти під користувачем якого не існує



Якщо натиснути на LOGOUT

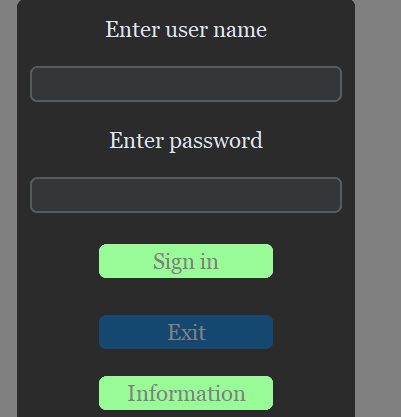


Перекине у вікно авторизації



Якщо натиснути Exit







Код файлу main.py

import tkinter

from tkinter import \*

import customtkinter as ctk

import db

import os

import re

class Window\_Log(ctk.CTk):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title("Window\_Log")

        self.geometry('400x400')

        self.configure(fg\_color='gray')

        self.resizable(False, False)

        self.count = 0

        frame = ctk.CTkFrame(master=self)

        frame.place(relx=0.5, rely=0.5, anchor='center')

        ctk.CTkLabel(master=frame, font=("Georgia", 18), text="Enter user name").grid(row=0, column=0, columnspan=2, pady=10)

        self.log\_entry = ctk.CTkEntry(master=frame, width=250, height=30, font=("Georgia", 18))

        self.log\_entry.grid(row=1, column=0, columnspan=2, padx=10, pady=5)

        ctk.CTkLabel(master=frame, font=("Georgia", 18), text="Enter password").grid(row=2, column=0, columnspan=2, pady=10)

        self.pwd\_entry = ctk.CTkEntry(master=frame, width=250, height=30, font=("Georgia", 18), show='\*')

        self.pwd\_entry.grid(row=3, column=0, columnspan=2, padx=10, pady=5)

        self.log\_b = ctk.CTkButton(master=frame, font=("Georgia", 18), text="Sign in",text\_color="gray", fg\_color="palegreen", command=self.sing\_in)

        self.log\_b.grid(row=4, column=0, columnspan=2, pady=18)

        self.exit\_b = ctk.CTkButton(master=frame, font=("Georgia", 18), text="Exit", text\_color="gray", fg\_color="palegreen", command=self.exit\_command)

        self.exit\_b.grid(row=5, column=0, columnspan=2, pady=10)

        self.info = ctk.CTkButton(master=frame, font=("Georgia", 18), text="Information", text\_color="gray", fg\_color="palegreen", command=self.info\_command)

        self.info.grid(row=6, column=0, columnspan=2, pady=10)

    # Кнопка виходу

    def exit\_command(self):

        self.destroy()

        return 0

    # Кнопка довідка

    def info\_command(self):

        tkinter.messagebox.showinfo(title="Information", message=f"""

                                    Creator:

        Muzychka-Skrypka Oleksandra FB-04,

                                    Variant: 10

        The presence of ordinary and capital letters, as well as signs of

        arithmetic operations.""")

    # Кнопка авторизації

    def sing\_in(self):

        login =  self.log\_entry.get()

        passwd  = self.pwd\_entry.get()

        if login != "":

            try:

                User = DataBase.data\_user[login]

            except KeyError:

                tkinter.messagebox.showerror(title= "Error", icon="error", message=f"'{login}' don't registrated")

            else:

                if User != None and User['pwd'] != passwd:

                    if self.count == 2:

                        tkinter.messagebox.showwarning(title="Warning", icon="warning", message=f"Stupid hacker 💩")

                        self.exit\_command()

                    else:

                        tkinter.messagebox.showwarning(title="Warning", icon="warning", message=f"Password is not correct")

                    self.count = self.count + 1

                elif User['ban'] == True:

                    tkinter.messagebox.showwarning(title="Warning", icon="warning", message=f"'{login}' was banned. What did you do?....")

                else:

                    self.exit\_command()

                    CtrlPan = ControlPanel(login)

                    CtrlPan.mainloop()

        else:

            tkinter.messagebox.showerror(title="Error warning", icon="error", message=f"""

Enter login please, then click 'Sign in'""")

class ControlPanel(ctk.CTk):

    def \_\_init\_\_(self, login):

        super().\_\_init\_\_()

        self.user = login

        self.title("Control")

        self.geometry('800x400')

        self.configure(fg\_color='black')

        self.resizable(False, False)

        self.apply\_butt = ctk.CTkButton(master=self, font = ("Georgia", 18), text = "Change password", text\_color = "gray", fg\_color = "palegreen", command = self.changeP)

        self.apply\_butt.grid(row=0, column=0, padx=10, pady=10)

        self.logout\_butt = ctk.CTkButton(master=self, font=("Georgia", 18), text="LOGOUT", text\_color="gray", fg\_color="lightcoral", command = self.log\_out)

        self.logout\_butt.grid(row=1, column=0, padx=10, pady=10)

        self.inf\_butt = ctk.CTkButton(master=self, font=("Georgia", 18), text="Information", text\_color="gray", fg\_color="palegreen", command = self.info\_command)

        self.inf\_butt.grid(row=2, column=0, padx=10, pady=10)

        if self.user == "ADMIN":

            self.geometry('800x400')

            self.add\_usr\_butt = ctk.CTkButton(master = self, text = "Sing up new user", font = ("Georgia", 18), text\_color = "gray", fg\_color = "palegreen", command = self.add\_usr)

            self.add\_usr\_butt.grid(row=0, column=1, padx=10, pady=10)

            self.new\_user = ctk.CTkEntry(master = self)

            self.new\_user.grid(row=0, column=2, padx=10, pady=10)

            self.add\_contr = ctk.CTkButton(master = self, text = "Add password control", font = ("Georgia", 18), text\_color = "gray", fg\_color = "palegreen", command = self.add\_pwd\_control)

            self.add\_contr.grid(row=1, column=1, padx=10, pady=10)

            self.re\_contr = ctk.CTkButton(master=self, text="Remove password control", font=("Georgia", 18), text\_color="gray", fg\_color="palegreen", command = self.re\_pwd\_control)

            self.re\_contr.grid(row=1, column=2, padx=10, pady=10)

            self.disable\_butt = ctk.CTkButton(master=self, text="Ban user", font=("Georgia", 18), text\_color="gray", fg\_color="palegreen", command = self.Ban\_User)

            self.disable\_butt.grid(row=2, column=1, padx=10, pady=10)

            self.able\_butt = ctk.CTkButton(master=self, text="Unban user", font=("Georgia", 18), text\_color="gray", fg\_color="palegreen", command = self.Unban\_User)

            self.able\_butt.grid(row=2, column=2, padx=10, pady=10)

            self.usersList = tkinter.Listbox(master=self, background="lightgray", selectmode="multiple")

            self.usersList.grid(row=3, column=0, columnspan=3, padx=60, pady=10, sticky="ew")

            self.Users\_List()

    def changeP(self):

        CtrlPass = Change\_PASS(self.user)

        CtrlPass.mainloop()

    def log\_out(self):

        self.destroy()

        logWin = Window\_Log()

        logWin.mainloop()

    # Кнопка довідка

    def info\_command(self):

        tkinter.messagebox.showinfo(title="Information", message=f"""

                                    Creator:

        Muzychka-Skrypka Oleksandra FB-04,

                                    Variant: 10

        The presence of ordinary and capital letters, as well as signs of

        arithmetic operations.""")

    def add\_usr(self):

        self.new\_usr = self.new\_user.get()

        if self.new\_usr:

            if self.new\_usr not in DataBase.data\_user:

                DataBase.AddUser(self.new\_usr)

            else: tkinter.messagebox.showinfo(title="Error", message="user already exists ")

        else: tkinter.messagebox.showinfo(title="Error", message="invalid username")

        self.Users\_List()

    def Users\_List(self):

        self.usersList.delete(0,END)

        for index,username in enumerate(DataBase.data\_user):

            user = DataBase.data\_user[username]

            self.usersList.insert(index,f"{username}    nocontrol "if DataBase.data\_user[username]["restrictions"] == False else f"{username }    control")

            color = "violet" if DataBase.data\_user[username]["ban"] == True else None

            self.usersList.itemconfig(index,bg=color)

    def add\_pwd\_control(self):

        for user in self.usersList.curselection():

            target = self.usersList.get(user).split()[0]

            DataBase.Add\_Control(target)

        self.Users\_List()

    def re\_pwd\_control(self):

        for user in self.usersList.curselection():

            target = self.usersList.get(user).split()[0]

            DataBase.Re\_Control(target)

        self.Users\_List()

    def Unban\_User(self):

        for user in self.usersList.curselection():

            target = self.usersList.get(user).split()[0]

            DataBase.UnbanUser(target)

        self.Users\_List()

    def Ban\_User(self):

        for user in self.usersList.curselection():

            target = self.usersList.get(user).split()[0]

            DataBase.BanUser(target)

        self.Users\_List()

class Change\_PASS(ctk.CTk):

    def \_\_init\_\_(self, login):

        super().\_\_init\_\_()

        self.user = login

        self.title("Contol password")

        self.geometry('500x400')

        self.configure(fg\_color='black')

        self.resizable(False, False)

        ctk.CTkLabel(master=self, font=("Georgia", 18), text="Old Password").pack()

        self.old\_pwd = ctk.CTkEntry(master=self, width=250)

        self.old\_pwd.pack(ipadx=2, ipady=2)

        ctk.CTkLabel(master=self, font=("Georgia", 18), text="New Password").pack()

        self.new\_pwd = ctk.CTkEntry(master=self, width=250)

        self.new\_pwd.pack(ipadx=2, ipady=2)

        ctk.CTkLabel(master=self, font=("Georgia", 18), text="Confirm Password").pack()

        self.confirm\_pwd = ctk.CTkEntry(master=self, width=250)

        self.confirm\_pwd.pack(ipadx=2, ipady=2)

        self.apply\_button = ctk.CTkButton(master=self, font=("Georgia", 18), text="Change", text\_color="gray", fg\_color="palegreen", command=self.password\_dat)

        self.apply\_button.pack(ipadx=2, ipady=2)

    def password\_dat(self):

        self.oldpasswd =  self.old\_pwd.get()

        self.newpasswd  = self.new\_pwd.get()

        self.confirmpasswd = self.confirm\_pwd.get()

        if self.newpasswd != self.confirmpasswd:

            tkinter.messagebox.showerror(title="Error", message=f"New Password and Confirm Password is diferent")

        else:

            if DataBase.data\_user[self.user]["restrictions"] ==  True:

                rigthtih = self.passwordValidation(self.newpasswd)

                if rigthtih == True and self.oldpasswd != self.newpasswd and self.oldpasswd == DataBase.data\_user[self.user]["pwd"]:

                    tkinter.messagebox.showinfo(title="Complete", message="Changed password successfully")

                    DataBase.changePassword(self.user, self.newpasswd)

                else:

                    tkinter.messagebox.showinfo(title="Error", message="new password is incorrect\n or simmilar with old password")

            else:

                if self.oldpasswd != self.newpasswd and self.oldpasswd == DataBase.data\_user[self.user]["pwd"]:

                    tkinter.messagebox.showinfo(title="Complete", message="Changed password successfully")

                    self.destroy()

                    DataBase.changePassword(self.user, self.newpasswd)

                else: tkinter.messagebox.showerror(title="Error", icon="error", message="new password is incorrect\n or simmilar with old password")

    def passwordValidation(self, Password):

        mathCharacters = set('+-=/%')

        if re.search('[A-Z]',Password) is None:

            tkinter.messagebox.showerror(title="Error", icon="error", message="Your Password must have at least 1 uppercase letter.")

            return False

        elif re.search('[a-z]',Password) is None:

            tkinter.messagebox.showerror(title="Error", icon="error", message="Your Password must have at least 1 lowercase letter.")

            return False

        elif not mathCharacters.intersection(Password):

            tkinter.messagebox.showerror(title="Error", icon="error", message="Your Password must have at least 1 mathCharacters")

            return False

        else:

            return True

DataBase = db.Data()

DataBase.file\_init()

logWin = Window\_Log()

logWin.mainloop()

Код файлу db.py який керував базою даних

import json

import os

class Data:

    def \_\_init\_\_(self):

        self.state = True

        # file\_path = f"{os.path.dirname(os.path.abspath(\_\_file\_\_))}/data\_user.json"

        if os.path.exists('data\_user.json') == False:

            # self.state = False

            self.state = False

            self.data\_user = {

                'ADMIN': {

                    'pwd': '',

                    'su': True,

                    'ban': False,

                    'restrictions': False

                }

            }

    def file\_init(self):

        if self.state == True:

            with open(f"data\_user.json", "r") as file:

                self.data\_user = json.load(file)

            print("File exist")

        else:

            with open(f"data\_user.json", "w") as createBase:

                json.dump(self.data\_user, createBase)

            self.state = True

            print("Creating new file")

    def changePassword(self, username, newpassword):

        self.data\_user[username]["pwd"] = newpassword

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

    def AddUser(self, username):

        self.data\_user[username] = {

                'pwd': '',

                'su': False,

                'ban': False,

                'restrictions': False

            }

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

    def Add\_Control(self, username):

        self.data\_user[username]["restrictions"] = True

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

    def Re\_Control(self, username):

        self.data\_user[username]["restrictions"] = False

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

    def BanUser(self,username):

        self.data\_user[username]["ban"] = True

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

    def UnbanUser(self, username):

        self.data\_user[username]["ban"] = False

        with open(f"data\_user.json", "w+") as w\_base:

            json.dump(self.data\_user, w\_base)

Наповнення бд після тестування

{"ADMIN": {"pwd": "", "su": true, "ban": false, "restrictions": false}, "Test": {"pwd": "Test+", "su": false, "ban": false, "restrictions": true}, "Test2": {"pwd": "", "su": false, "ban": false, "restrictions": true}, "Test3": {"pwd": "", "su": false, "ban": false, "restrictions": false}, "Test4": {"pwd": "", "su": false, "ban": false, "restrictions": false}}