

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

Ім'я	Дата змінення
__pycache__	05.06.2023 23:29
db.py	05.06.2023 23:29
lab1_zpz.docx	05.06.2023 23:12
main.py	05.06.2023 23:27

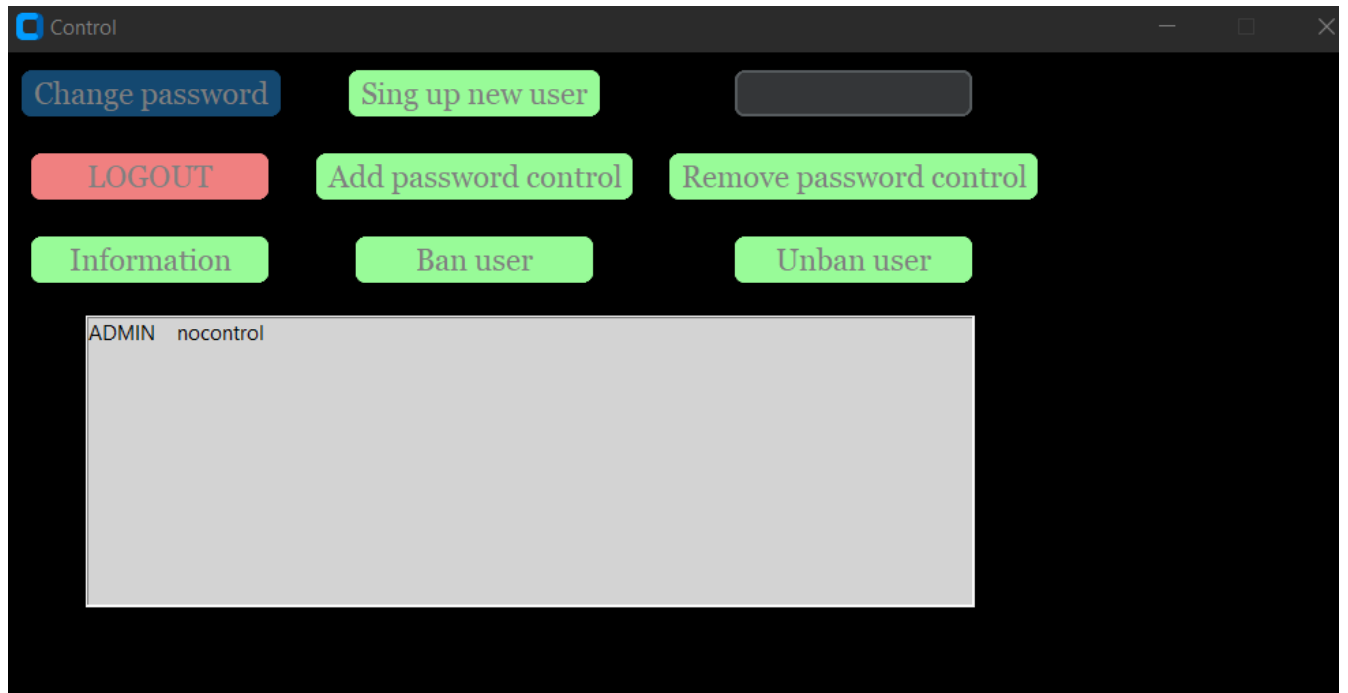
```
( after script)  
PS C:\Study\6_semestr\zpz\lab1> & C:/Users/twinc/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Study/6_seme  
Creating new file
```

```
{ } data_user.json > ...  
1  {"ADMIN": {"pwd": "", "su": true, "ban": false, "restrictions": false}}
```

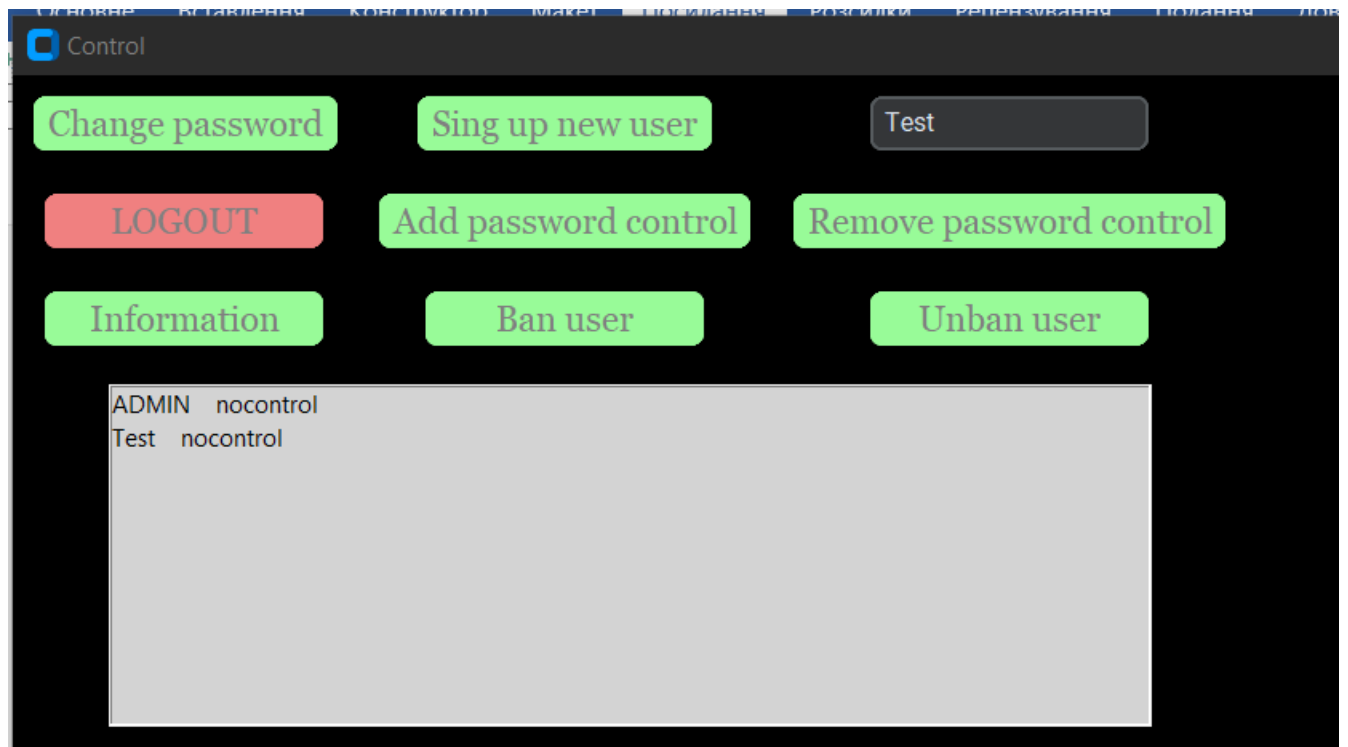
При повторному підключенні

```
Creating new file  
PS C:\Study\6_semestr\zpz\lab1> & C:/Users/twinc/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Study/6_semestr/zpz/lab1/my_lab1/main.py  
File exist
```

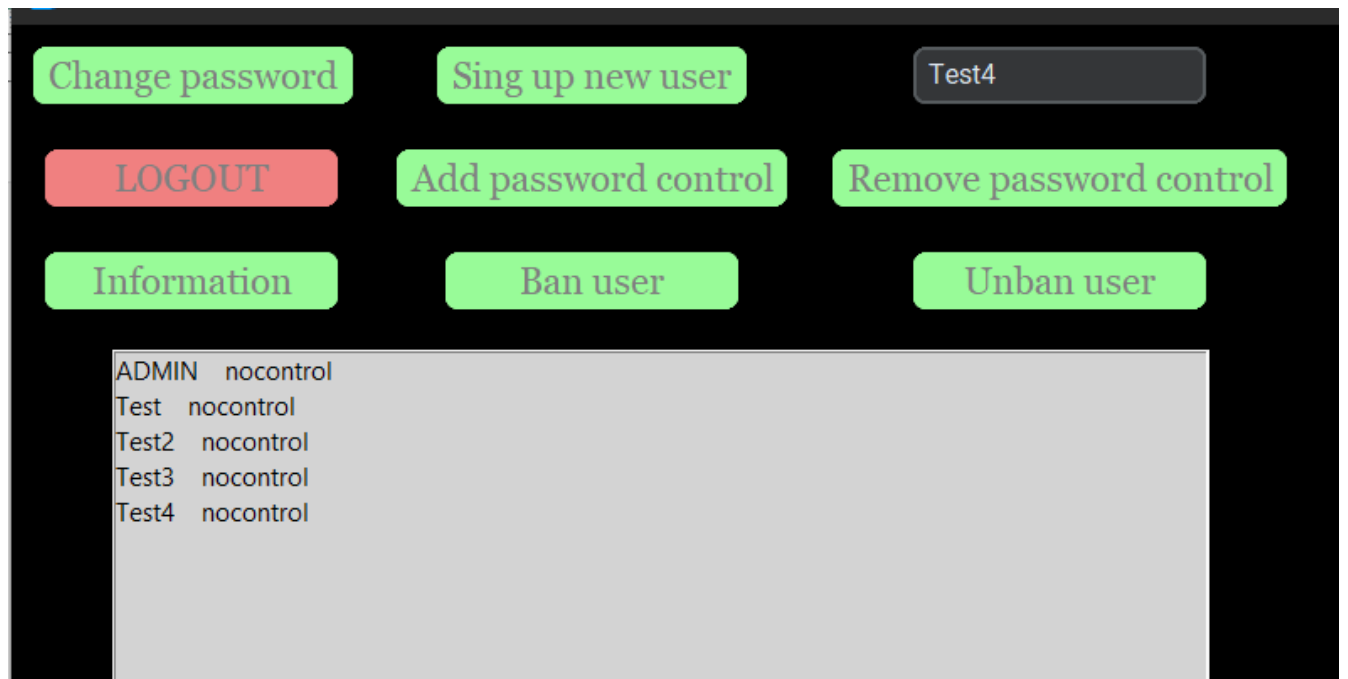
Режим Адміністратора



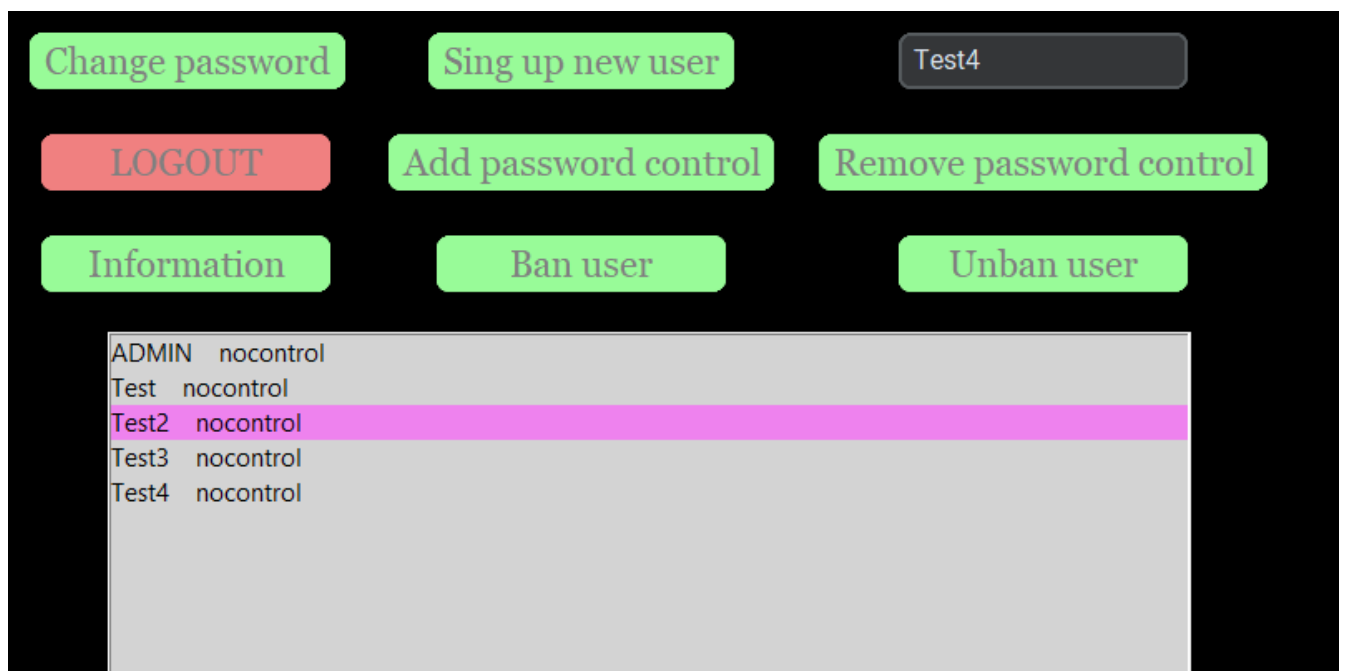
Додамо користувача Test



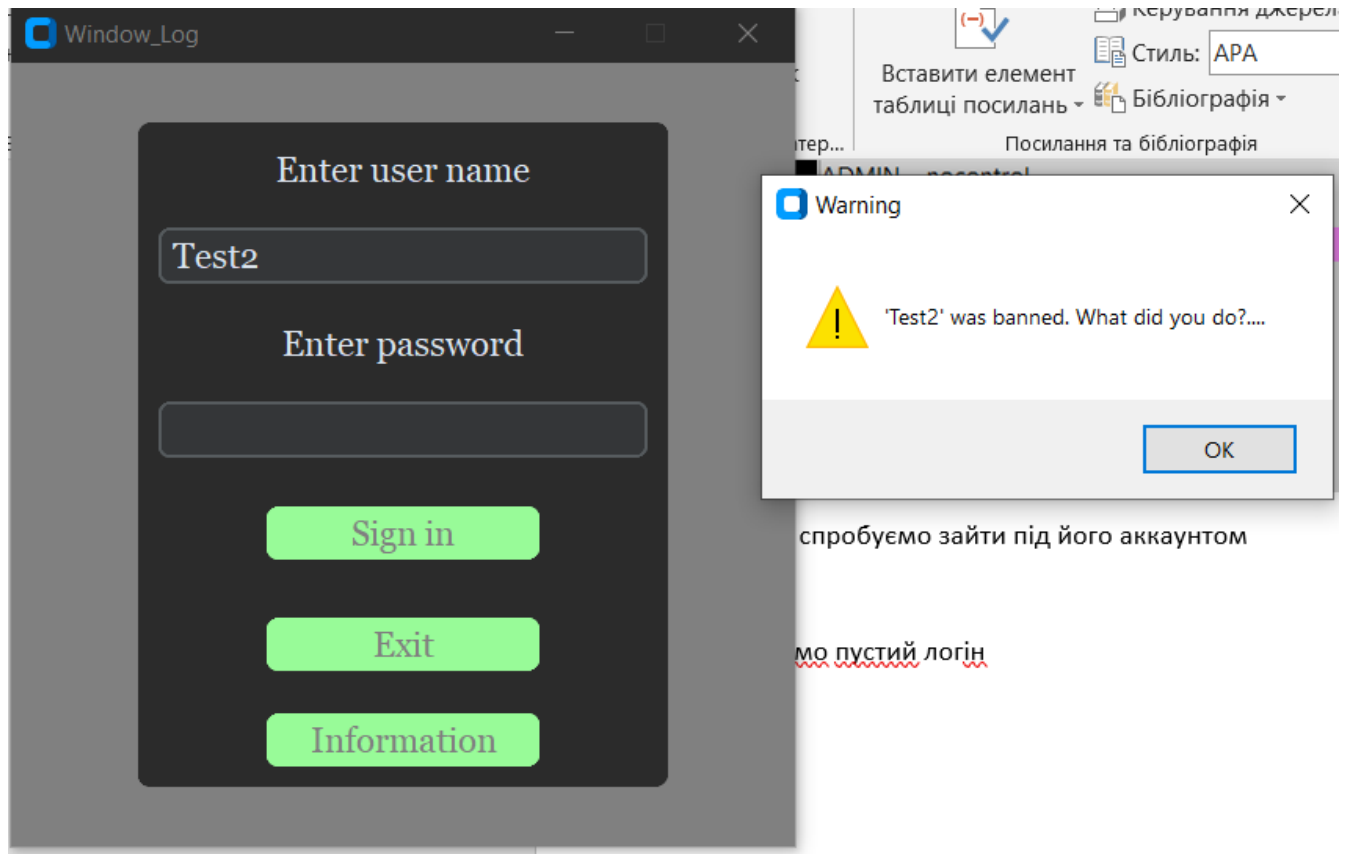
І ще декілька



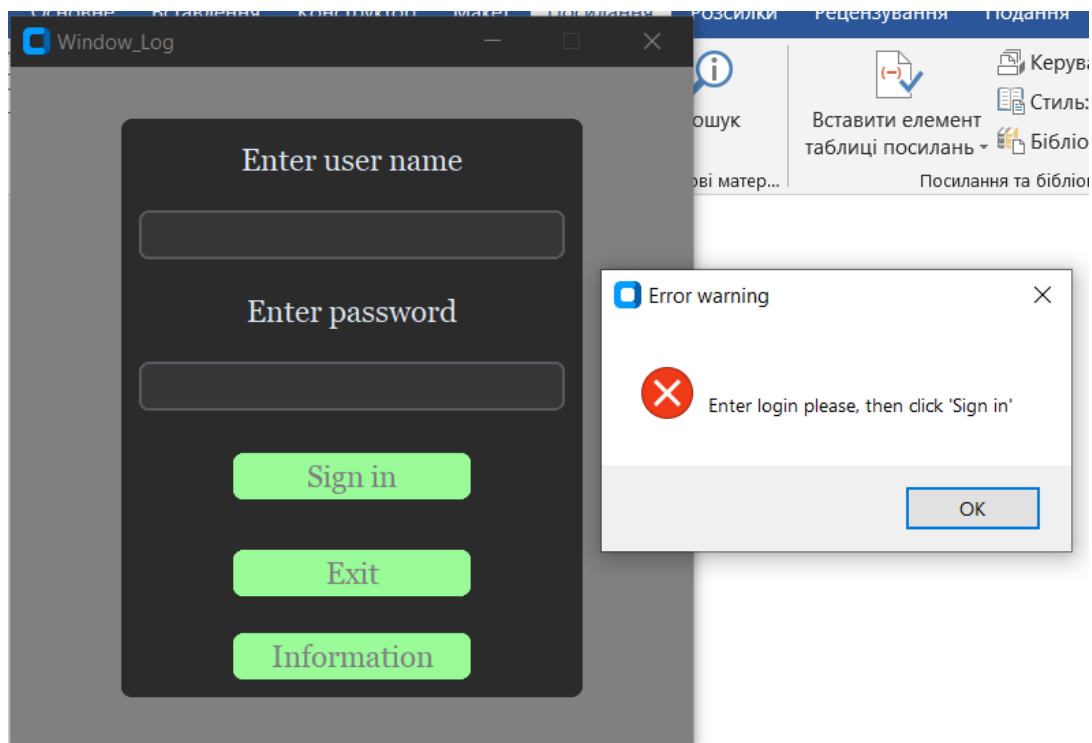
Заблокуємо Test2



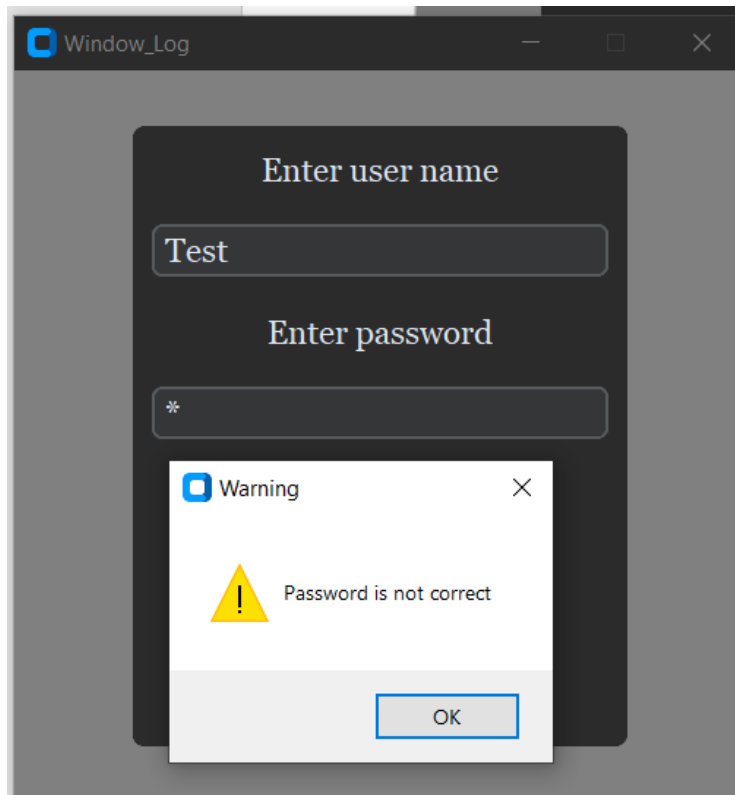
Тепер спробуємо зайти під його аккаунтом



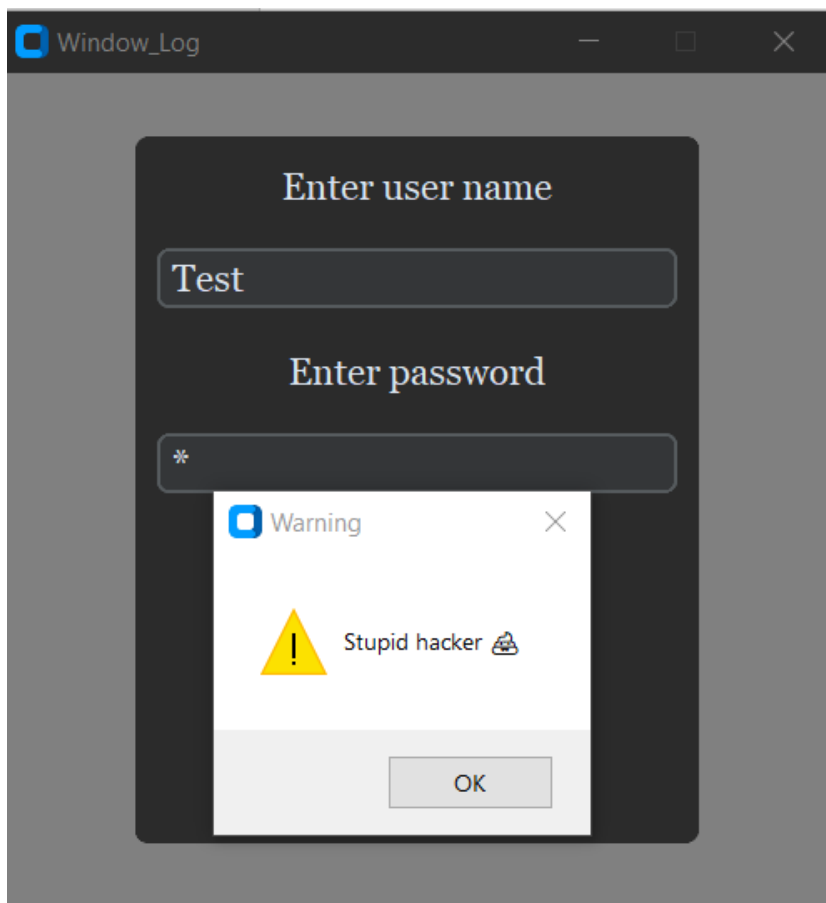
Введемо пустий логін



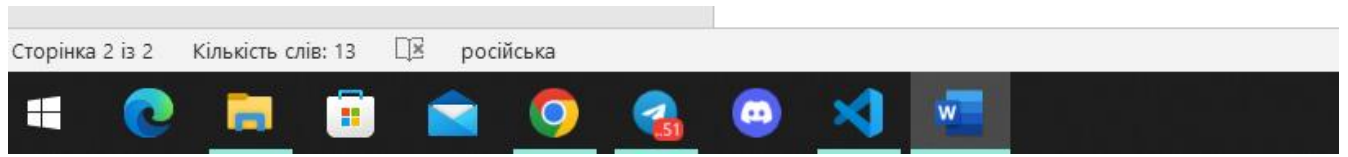
Неправильний пароль



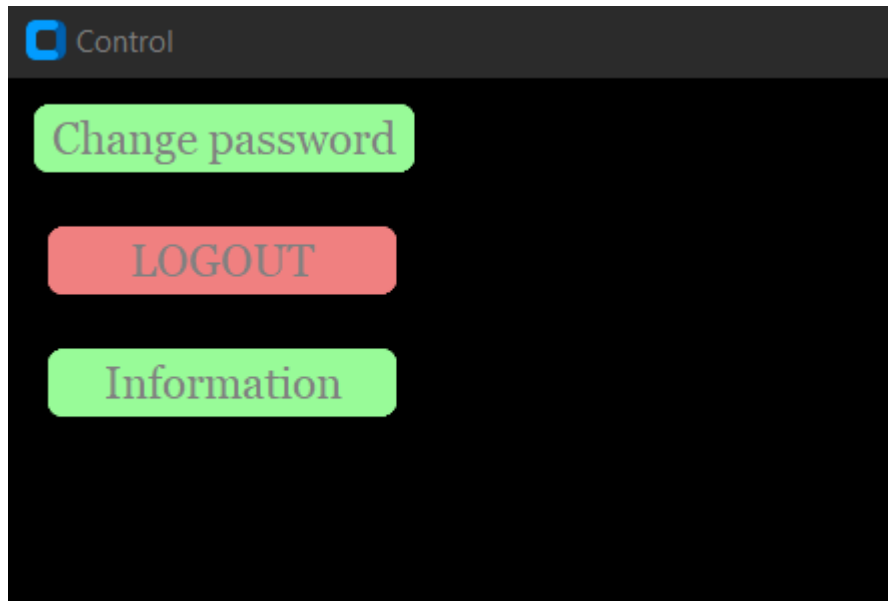
Якщо тричі неправильно ввести пароль



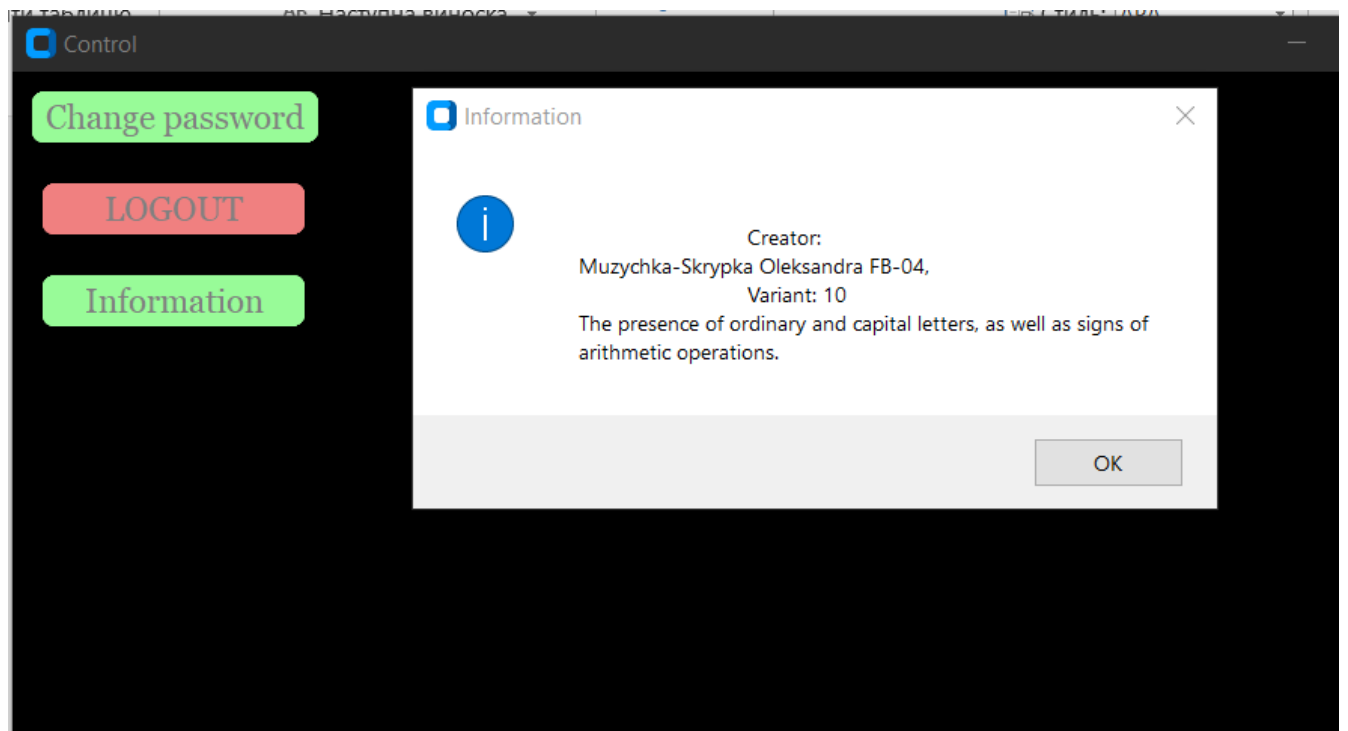
Потім програма закрилась



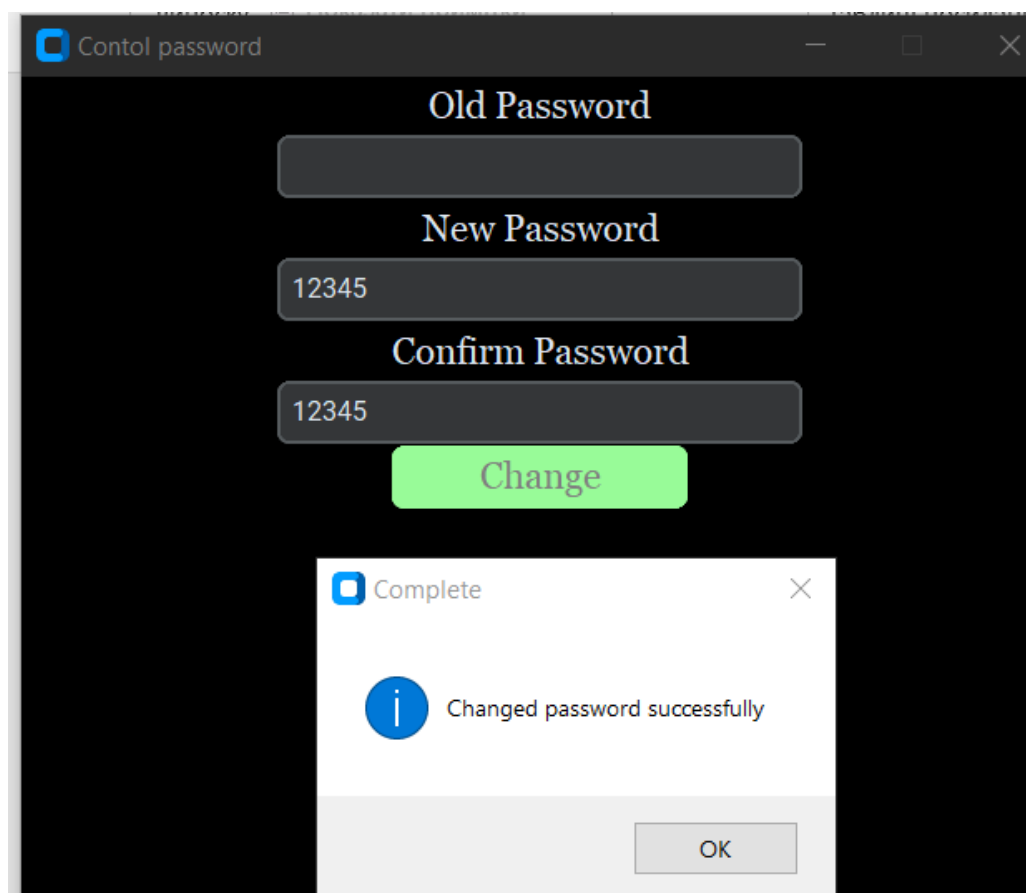
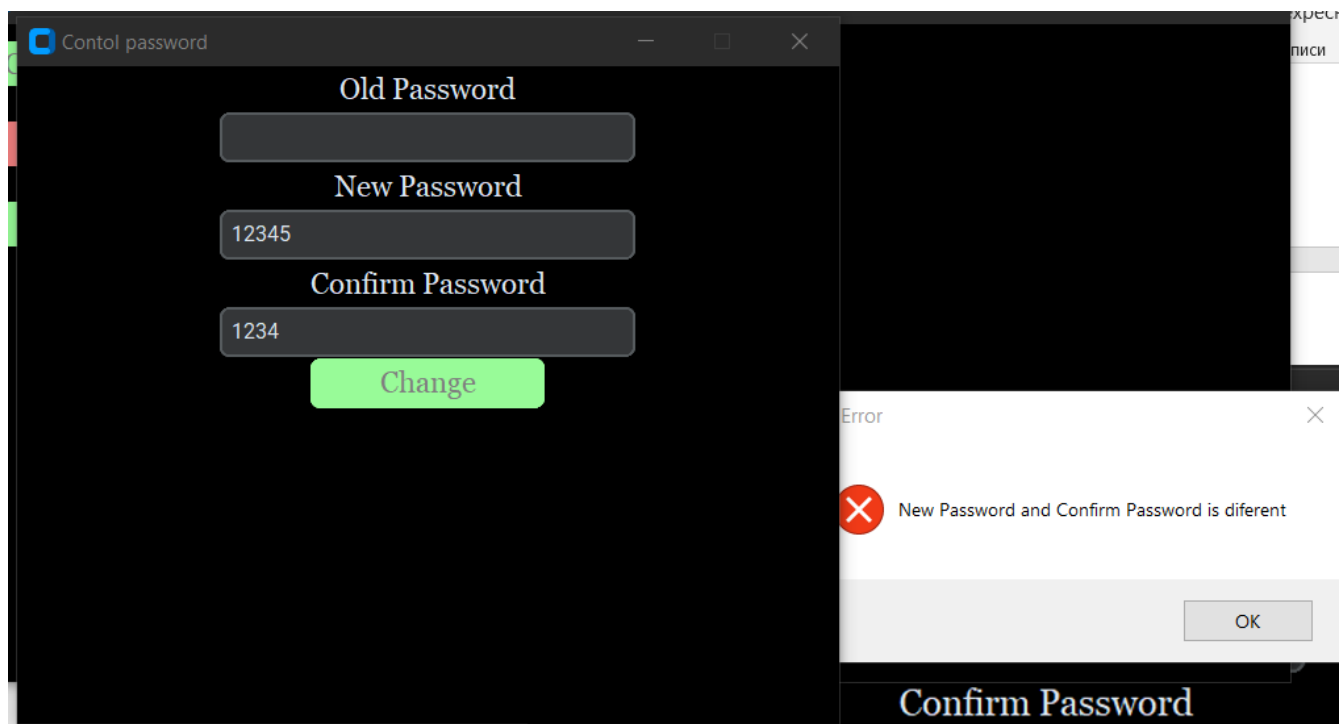
Режим користувача



Інформація



Спробуємо змінити пароль



Тепер спробуємо замінити на такий самий пароль

Contol password

Old Password

12345

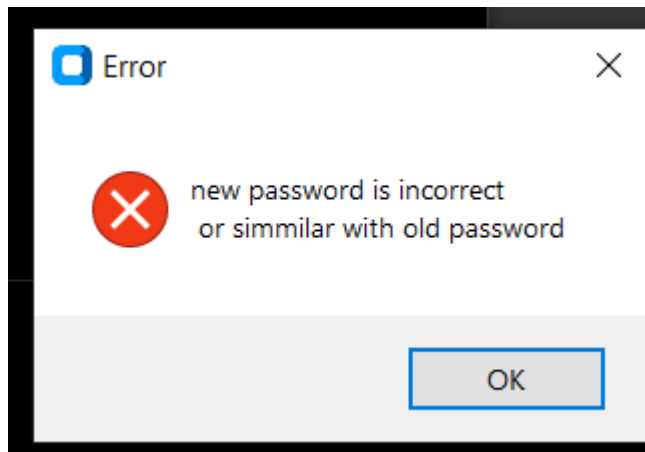
New Password

12345

Confirm Password

12345

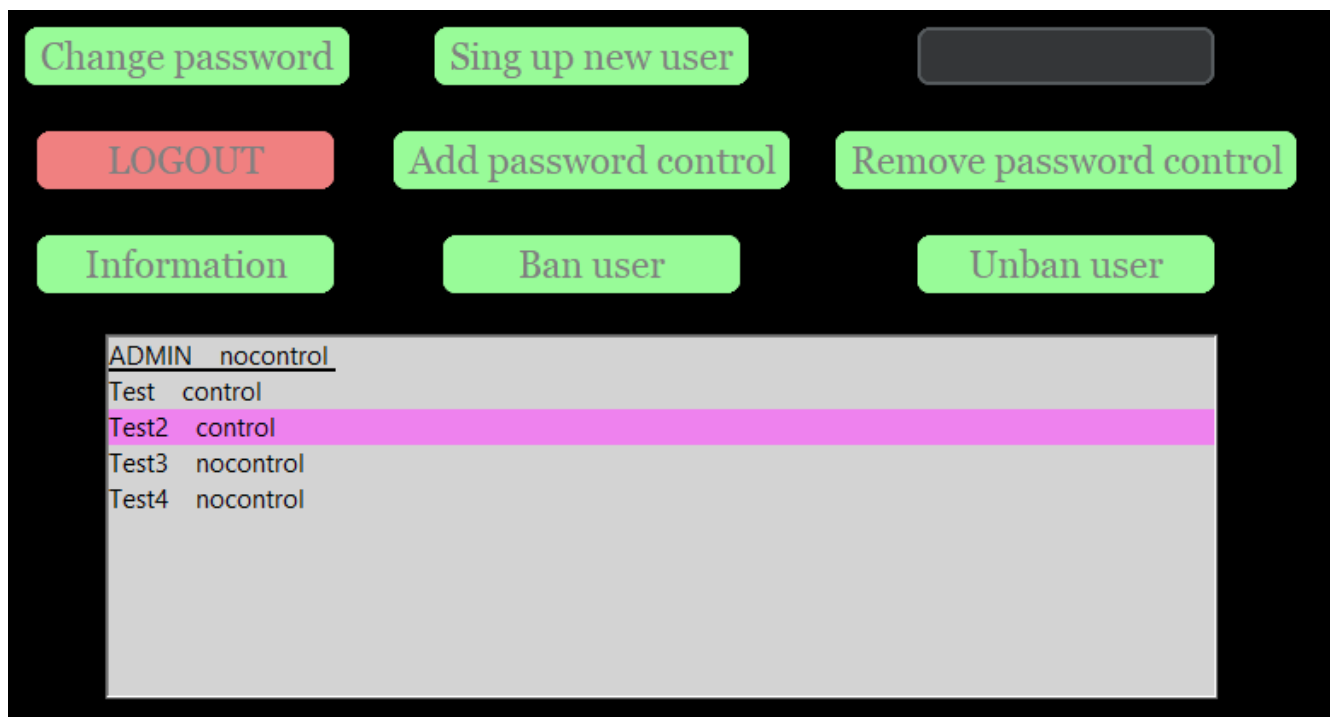
Change



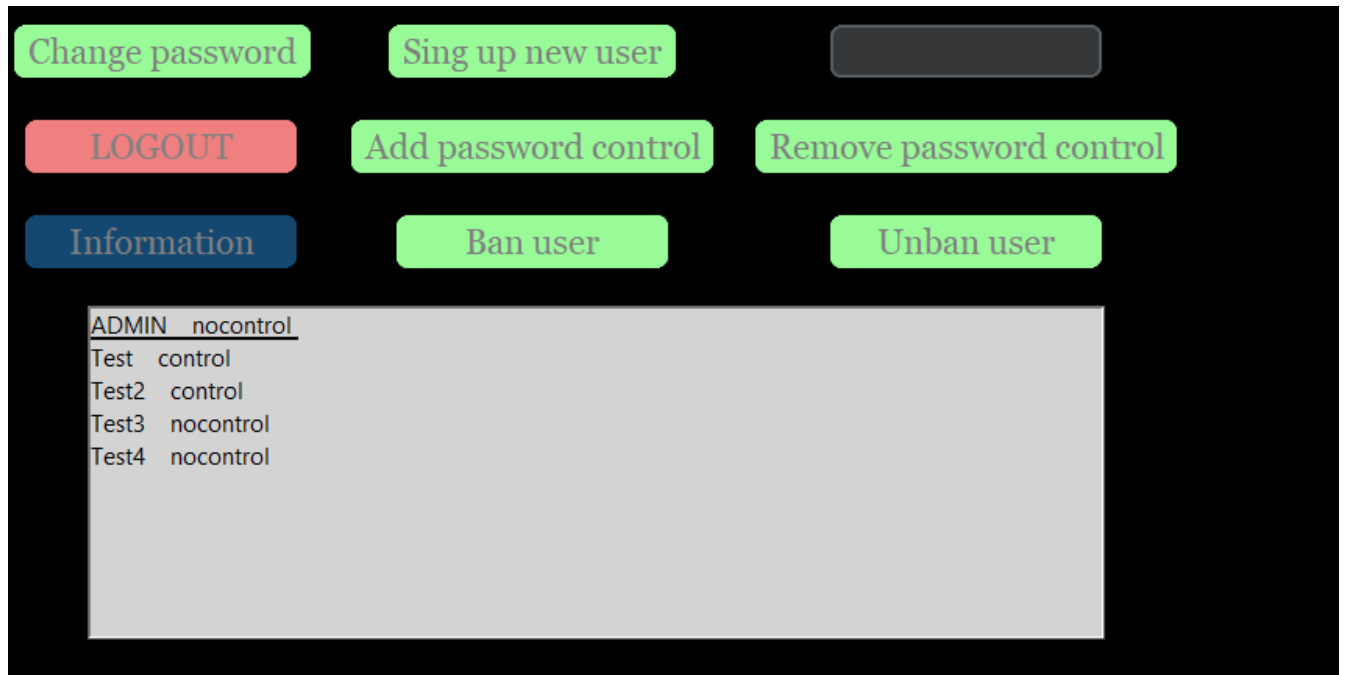
Повернемось режиму адміна та встановимо обмеження щодо пароллю для юзера Test та Test3



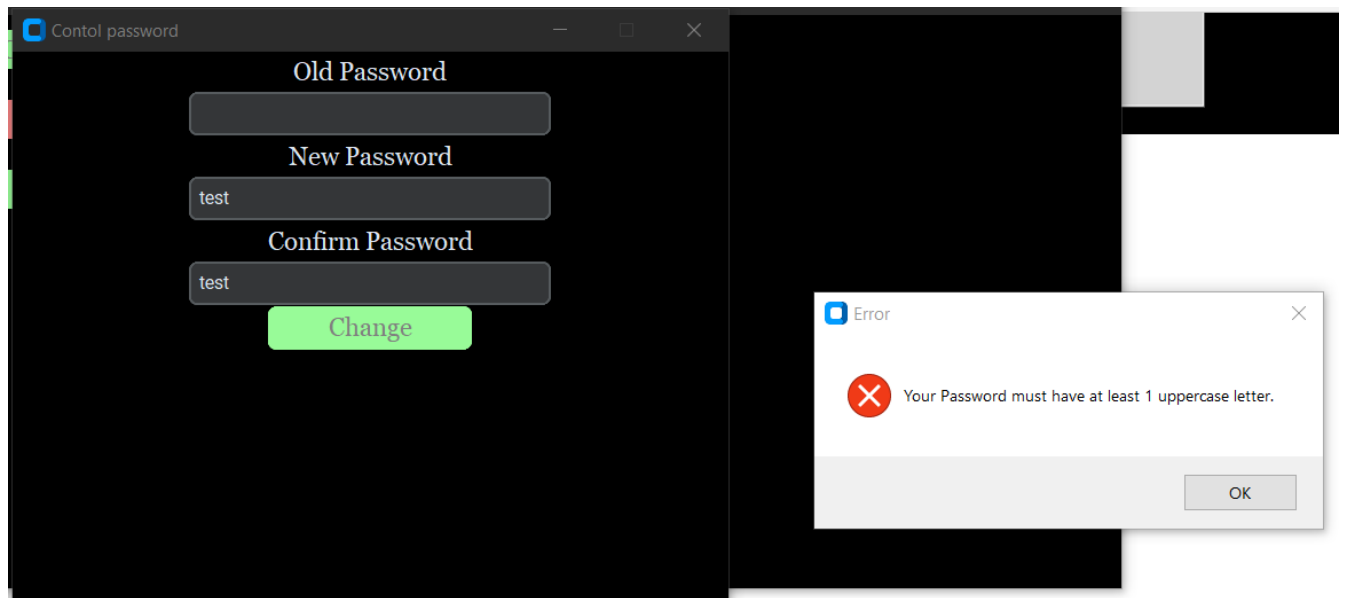
Видалимо обмеження для Test3

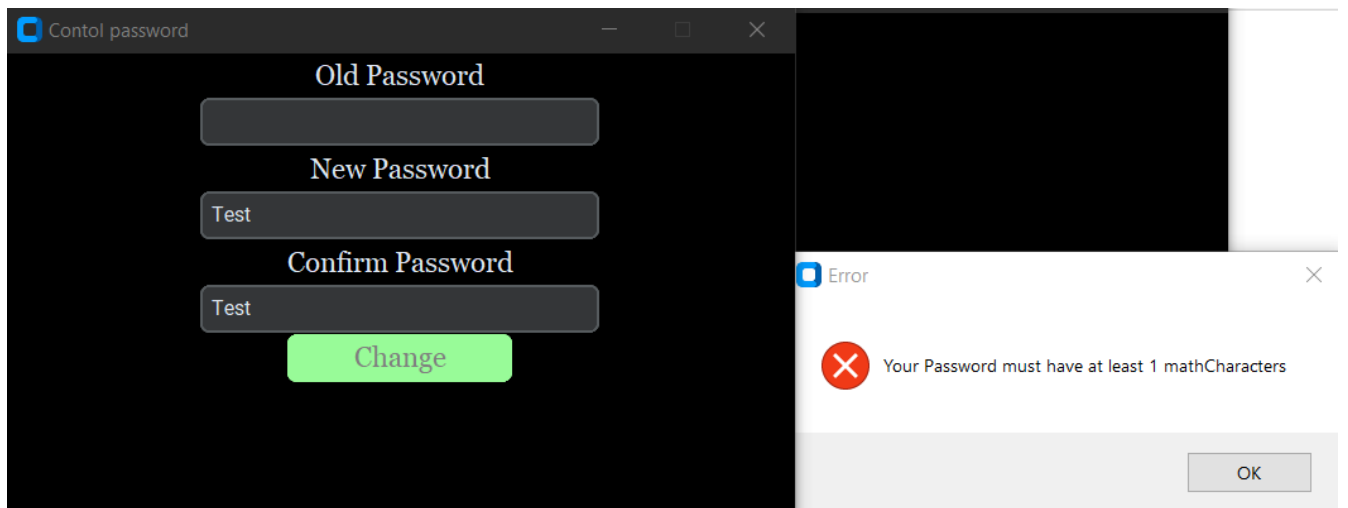
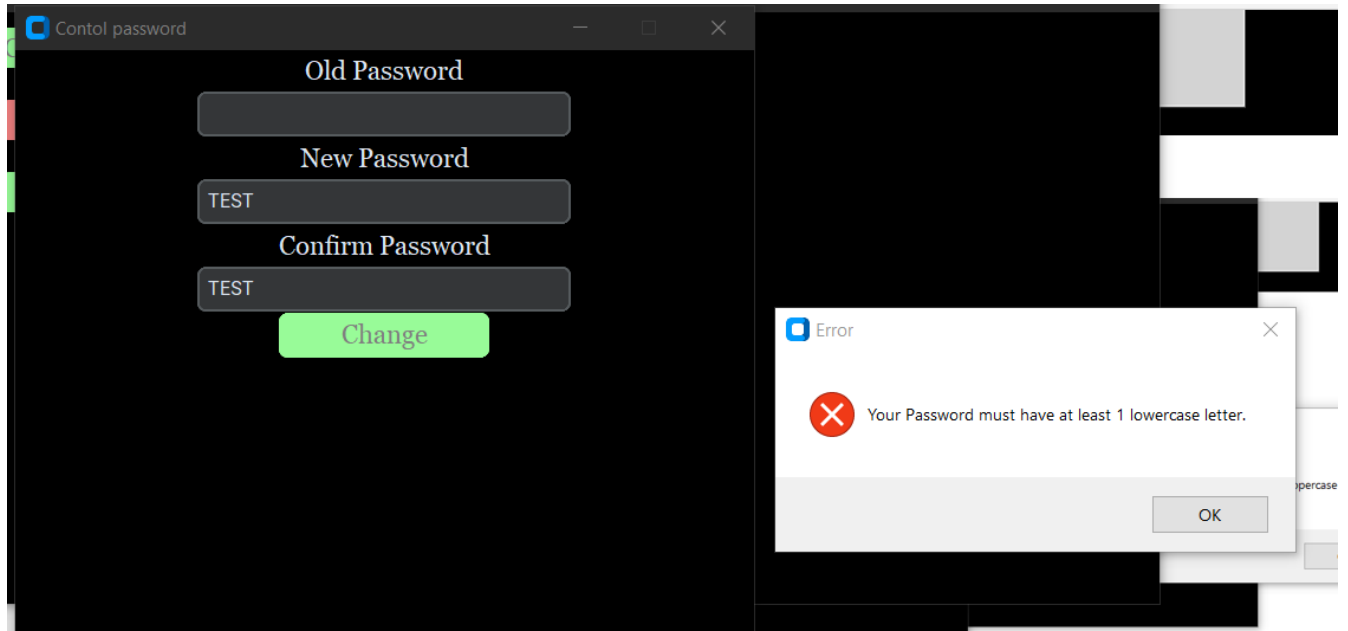


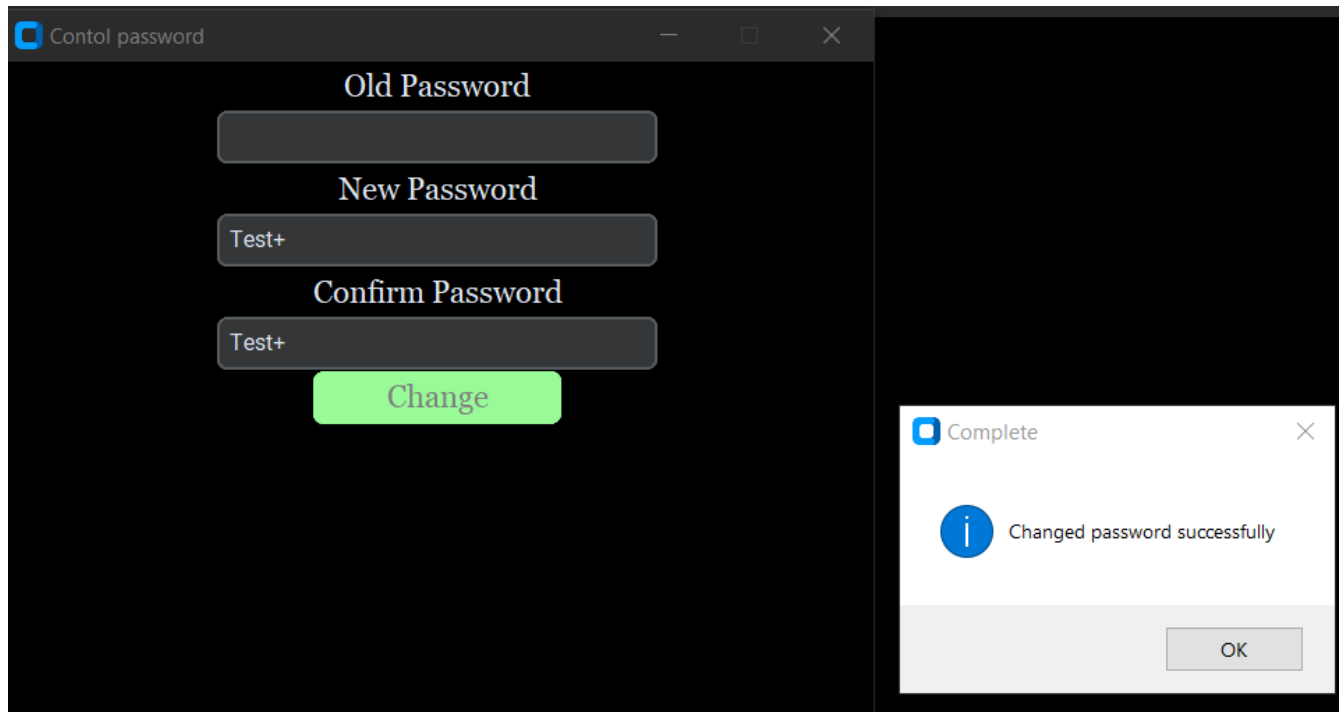
Розбанимо Test2



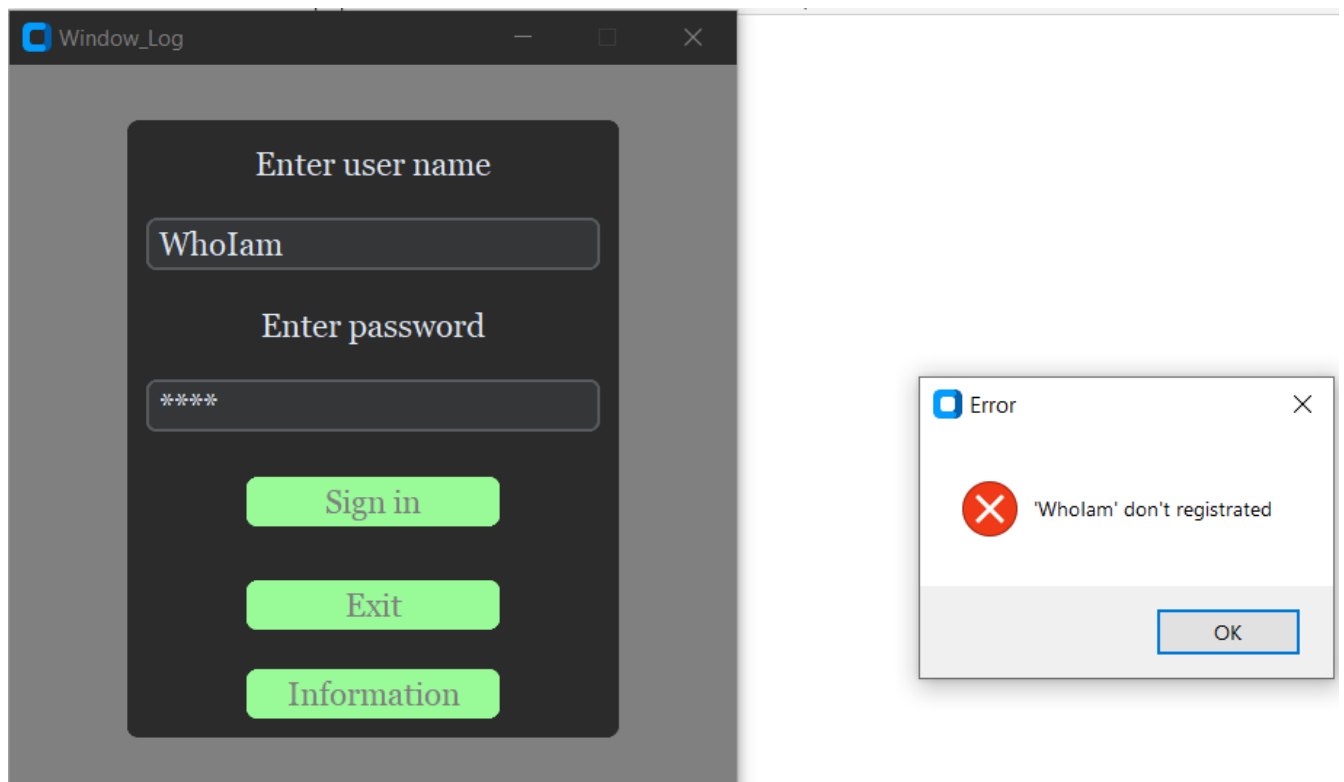
Перейдемо знову до користувача Test







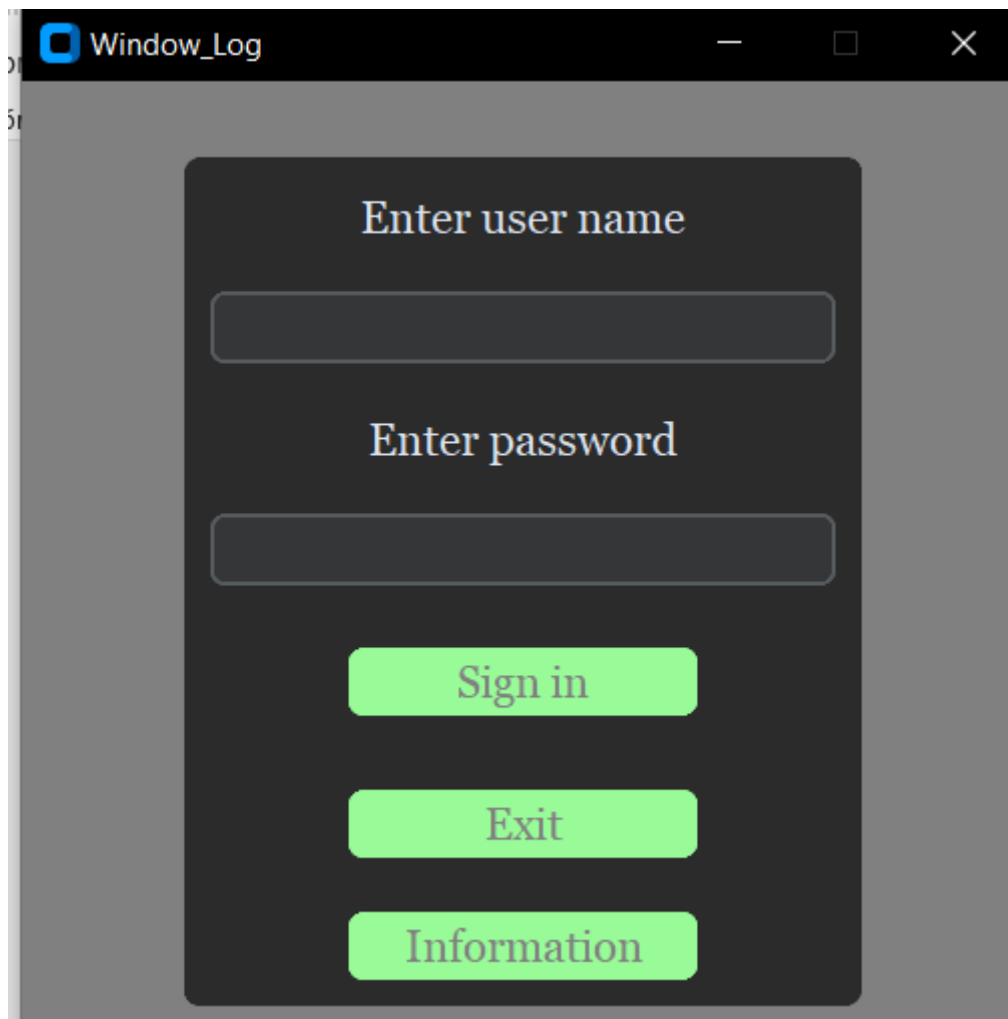
Спробуємо зайти під користувачем якого не існує



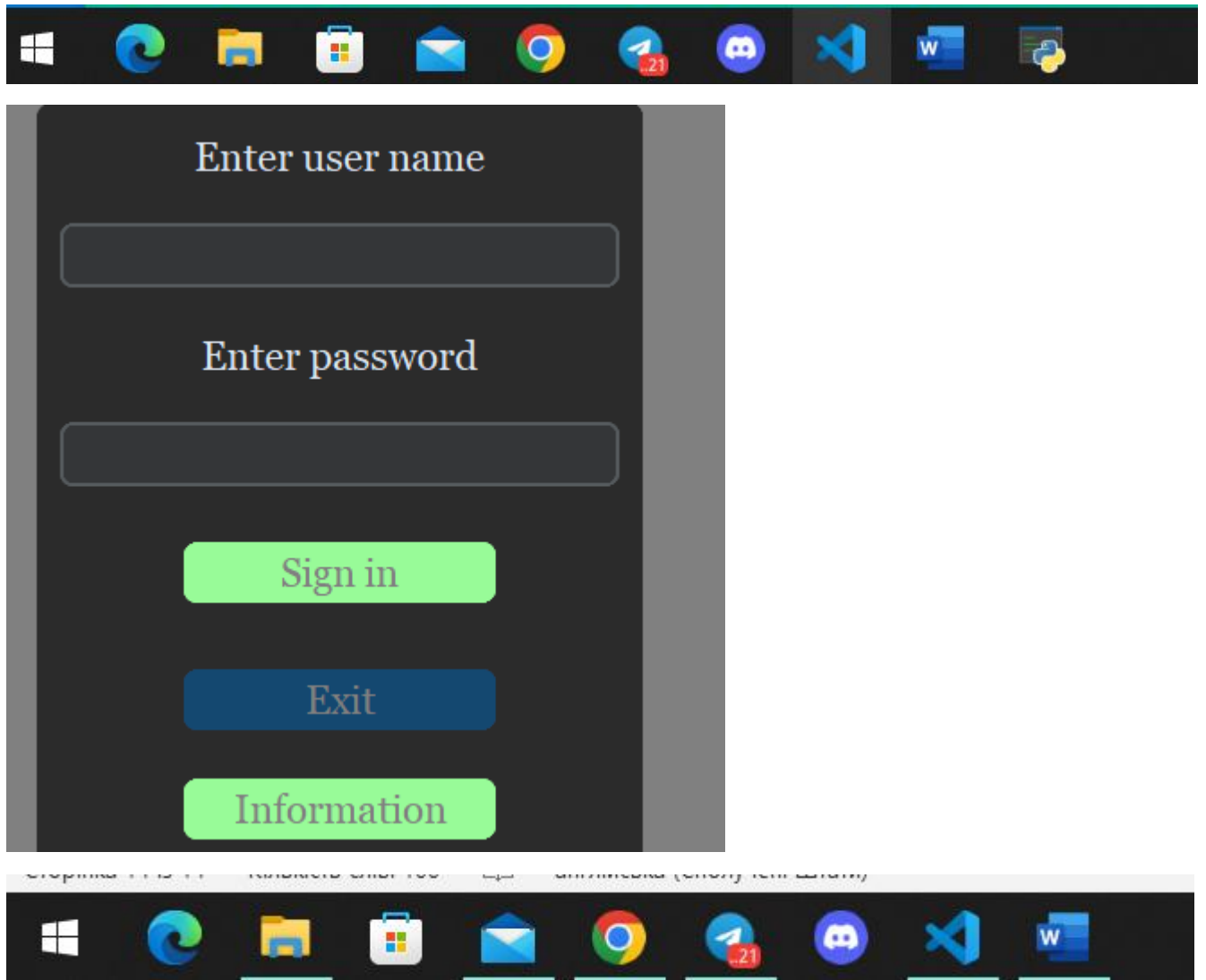
Якщо натиснути на LOGOUT



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(ctlk.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_but = ctlk.CTkButton(master=self, font = ("Georgia", 18), text =
"Change password", text_color = "gray", fg_color = "palegreen", command =
self.changeP)
        self.apply_but.grid(row=0, column=0, padx=10, pady=10)

        self.logout_but = ctlk.CTkButton(master=self, font=("Georgia", 18),
text="LOGOUT", text_color="gray", fg_color="lightcoral", command = self.log_out)
        self.logout_but.grid(row=1, column=0, padx=10, pady=10)

        self.inf_but = ctlk.CTkButton(master=self, font=("Georgia", 18),
text="Information", text_color="gray", fg_color="palegreen", command =
self.info_command)
        self.inf_but.grid(row=2, column=0, padx=10, pady=10)

        if self.user == "ADMIN":
            self.geometry('800x400')

```



```

        self.add_usr_but = 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_but.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_but = ctk.CTkButton(master=self, text="Ban user",
font=("Georgia", 18), text_color="gray", fg_color="palegreen", command =
self.Ban_User)
        self.disable_but.grid(row=2, column=1, padx=10, pady=10)

        self.able_but = ctk.CTkButton(master=self, text="Unban user",
font=("Georgia", 18), text_color="gray", fg_color="palegreen", command =
self.Unban_User)
        self.able_but.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:
                righthih = self.passwordValidation(self.newpasswd)
                if righthih == 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}}
```