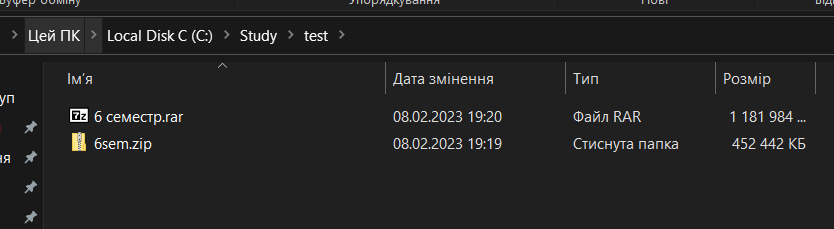
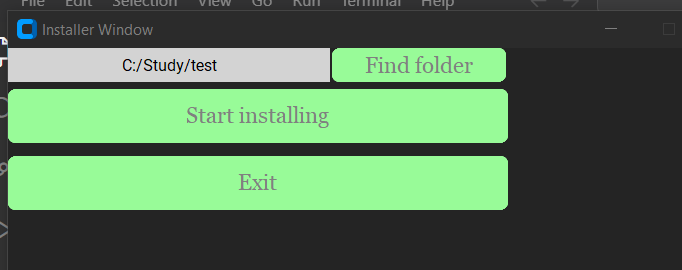
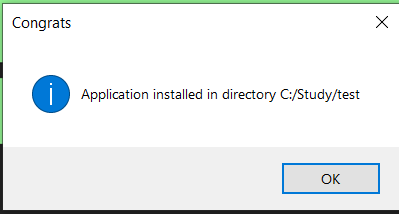
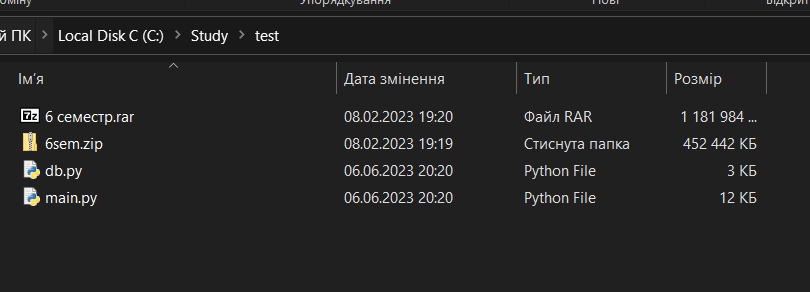


Оберемо папку

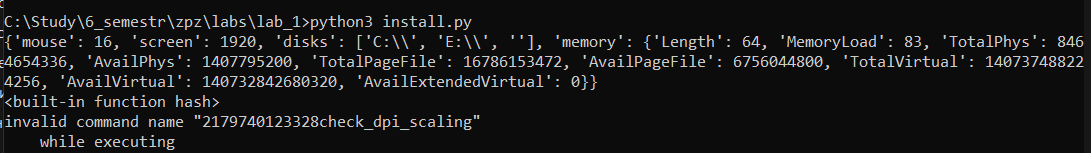




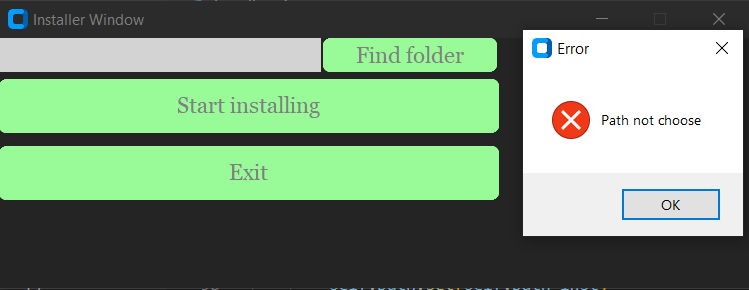




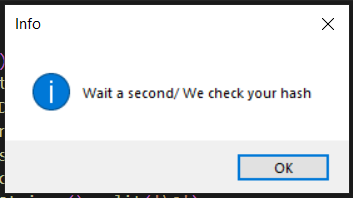
Вивела зібрані дані для наочності

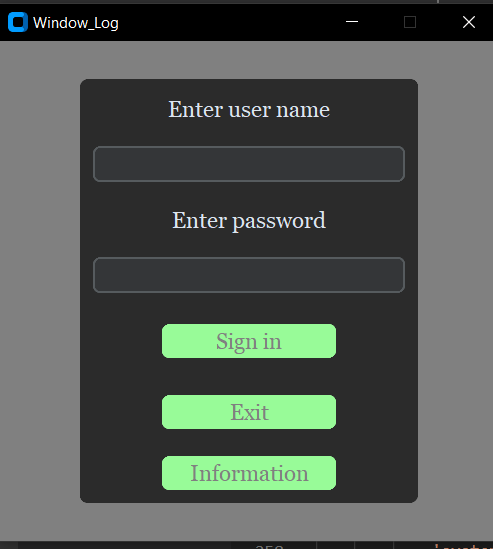


Якщо не обрати папку



Тепер запустимо змінений код до 1шої лаби





Install.py

import zipfile

import hashlib

import json

import win32api

import tkinter

import winreg

from tkinter import filedialog,messagebox

import customtkinter as ctk

import win32.lib.win32con as win32con

import platform

import pyautogui

APP = "AppLab2.zip"

class INSTALLER(ctk.CTk):

    screen\_width, \_ = pyautogui.size()

    print(screen\_width)

    def GetInfoUser(self):

        info = {

        'username': win32api.GetUserName(),

        'computername': win32api.GetComputerName(),

        'windowsdir': win32api.GetWindowsDirectory(),

        'systemdir': win32api.GetSystemDirectory(),

        'mouse': win32api.GetSystemMetrics(win32con.SM\_CMOUSEBUTTONS),

        'screen': win32api.GetSystemMetrics(win32con.SM\_CXSCREEN),

        'disks': win32api.GetLogicalDriveStrings().split('\0'),

        'memory': win32api.GlobalMemoryStatusEx()[0]

        }

        return info

    def setRegKey(self, value):

        # key = winreg.CreateKey(winreg.HKEY\_CURRENT\_USER, "SOFTWARE\\MuzychkaSkrypka")

        # winreg.SetValueEx(key, sign, 0, winreg.REG\_SZ,value)

        key = winreg.CreateKey(winreg.HKEY\_CURRENT\_USER, "SOFTWARE\\MuzychkaSkrypka")

        winreg.SetValueEx(key, "HashValue", 0, winreg.REG\_SZ, value)

        winreg.CloseKey(key)

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title ('Installer Window')

        self.geometry("800x400")

        self.resizable(False,False)

        self.path = tkinter.StringVar()

        self.find\_folder = ctk.CTkButton(master=self, font=("Georgia", 18), text = "Find folder", text\_color="gray",  fg\_color="palegreen", command = self.ShowFolder)

        self.find\_folder.grid(row = 1, column =5)

        self.start\_install = ctk.CTkButton(master=self, font=("Georgia", 18), text="Start installing", text\_color="gray",  fg\_color="palegreen", command=self.Install,width=400)

        self.start\_install.grid(column = 4,ipady=10,columnspan=2,pady = 5)

        self.label\_path = ctk.CTkLabel(master = self, width=250, fg\_color="lightgray", textvariable= self.path, text\_color="black")

        self.label\_path.grid(row = 1,columnspan = 4, column =1,ipadx = 5)

        self.end\_button = ctk.CTkButton(master=self, font=("Georgia", 18), text="Exit", text\_color="gray",  fg\_color="palegreen", command=self.exit\_but, width=400)

        self.end\_button.grid(row = 4, column = 4,ipady=10,columnspan=2,pady = 5)

    def exit\_but(self):

        self.destroy()

    def ShowFolder(self):

        self.path\_inst = filedialog.askdirectory(title ="Select Folder")

        self.path.set(self.path\_inst)

    def Install(self):

        dir = self.path.get()

        if not dir:

            tkinter.messagebox.showerror(title="Error", icon = "error", message = "Path not choose")

            return

        info = self.GetInfoUser()

        # generated\_hash = hashlib.md5(json.dumps(info).encode()).hexdigest()

        print(info)

        print(hash)

        json\_info = json.dumps(info).encode('utf-8')

        # hash\_object = hashlib.sha256(json\_info)

        hash\_object = hashlib.md5(json\_info)

        hash\_hex = hash\_object.hexdigest()

        self.setRegKey(hash\_hex)

        # self.setRegKey("Signature", generated\_hash)

        zip = zipfile.ZipFile(APP)

        try:

            zip.extractall(dir)

            zip.close()

            self.destroy()

            messagebox.showinfo(title="Congrats", message = f"Application installed in directory {dir}")

        except Exception:

            tkinter.messagebox.showerror(title="Error",message ="Something went wrong")

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

    Installer = INSTALLER()

    Installer.mainloop()

Та додала код до main.py

class HashCheck():

        def \_\_init\_\_(self):

            tkinter.messagebox.showinfo(title="Info", message="Wait a second/ We check your hash")

        def GetInfoUser(self):

            info\_hash = {

                'username': win32api.GetUserName(),

                'computername': win32api.GetComputerName(),

                'windowsdir': win32api.GetWindowsDirectory(),

                'systemdir': win32api.GetSystemDirectory(),

                'mouse': win32api.GetSystemMetrics(win32con.SM\_CMOUSEBUTTONS),

                # 'screen': win32api.GetSystemMetrics(win32con.SM\_CXSCREEN),

                'disks': win32api.GetLogicalDriveStrings().split('\0'),

                # 'memory': win32api.GlobalMemoryStatusEx()[0]

            }

            return info\_hash

        def getRegKey(self, sign):

            key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,"SOFTWARE\\MuzychkaSkrypka")

            value = winreg.QueryValueEx(key,sign)[0]

            return value

        def Hash\_Check(self):

            info = self.GetInfoUser()

            print(info)

            json\_info = json.dumps(info).encode('utf-8')

            hash\_object = hashlib.md5(json\_info)

            hash\_now = hash\_object.hexdigest()

            # hash\_now = hashlib.md5(json.dumps(info).encode()).hexdigest()

            cool\_hash = self.getRegKey("HashValue")

            print(hash\_now)

            print(cool\_hash)

            if (hash\_now == cool\_hash):

                return True

            else:

                return False

hash\_TF = HashCheck()

hash\_TF.Hash\_Check()

if hash\_TF:

    DataBase = db.Data()

    DataBase.file\_init()

    logWin = Window\_Log()

    logWin.mainloop()

else:

  tkinter.messagebox.showinfo(title="Error", message="Your hash is incorrect")