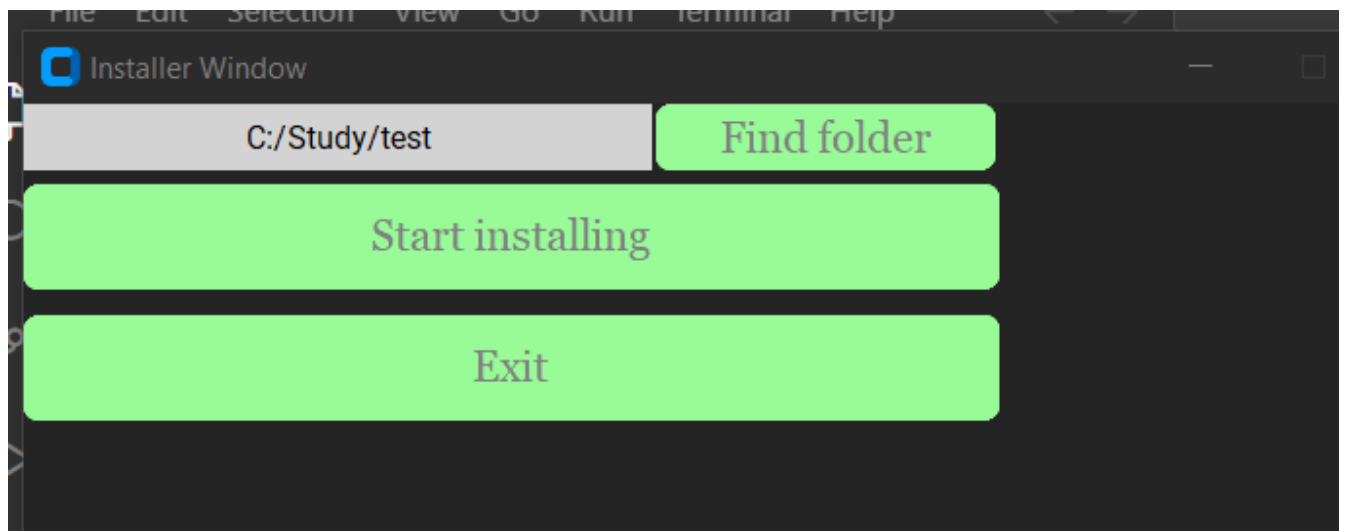
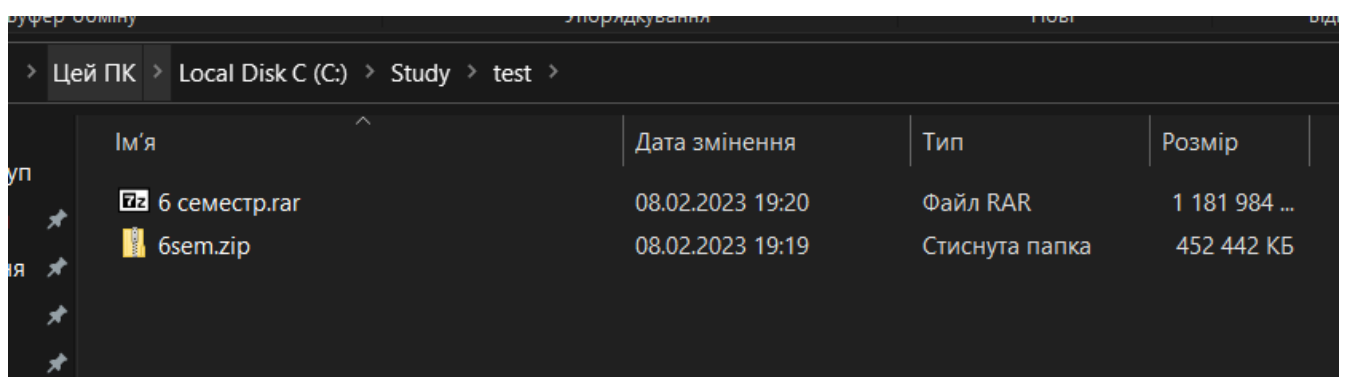
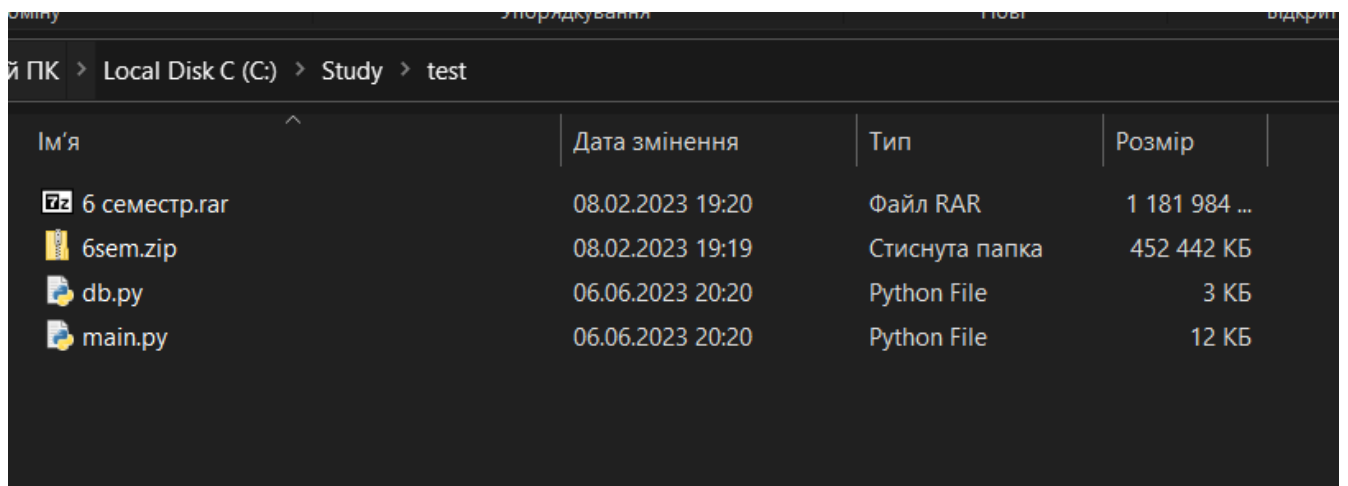
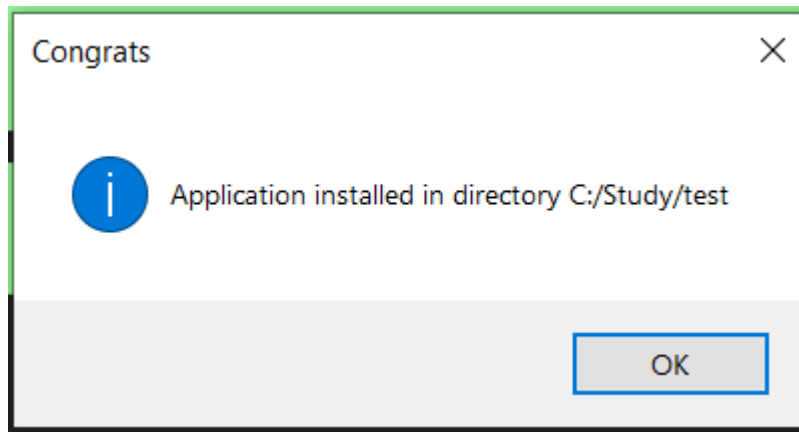
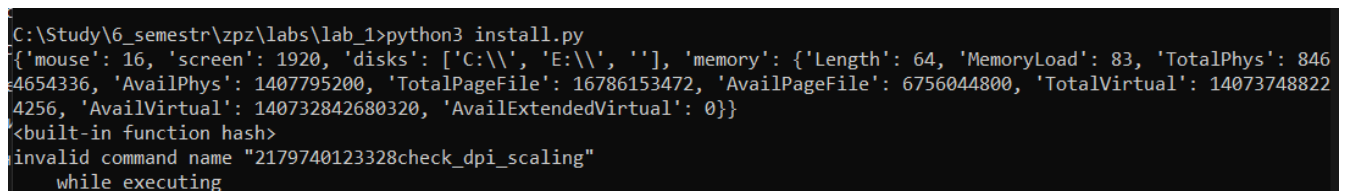


Оберемо папку

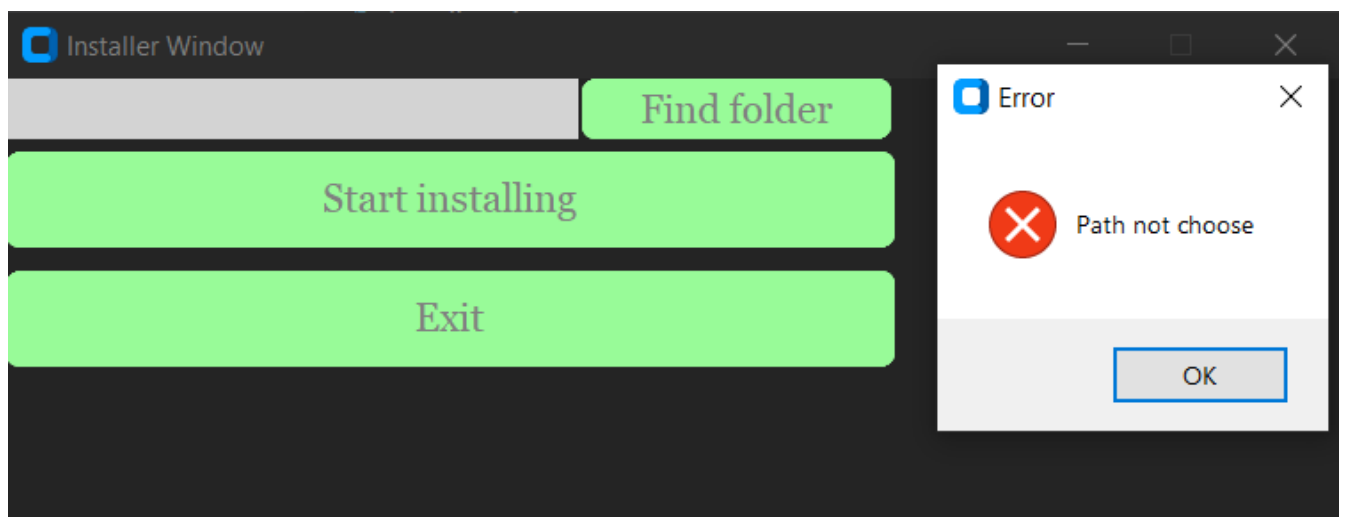




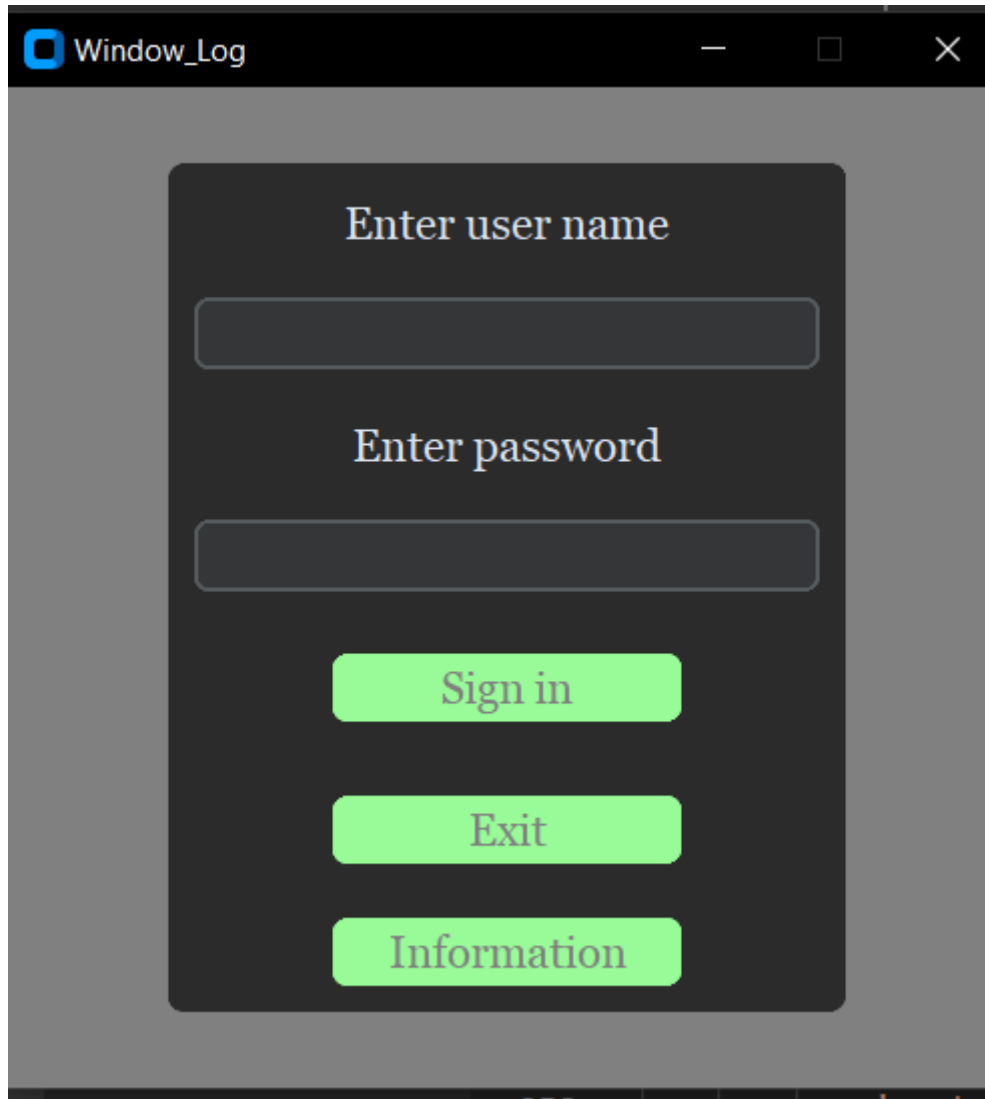
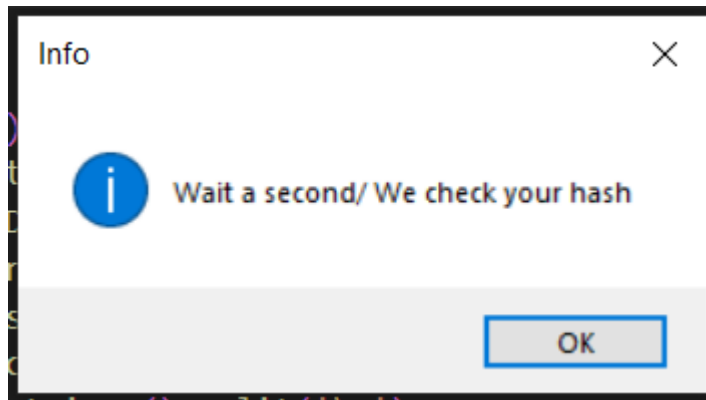
Вивела зібрані дані для наочності



Якщо не обрати папку



Тепер запусимо 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\\MuzychkaSkryпка")
    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")

```