

## שם התרגיל:

שמירת קבצים על השרת עם ממשק גרפי ללקוח

## מטרות:

- להבין עבודה עם קבצים מצד שרת.
- לתרגל תקשורת ללקוח-שרת (upload/download/delete) (PyQt/Tkinter).
- לתרגל עבודה עם ממשק גרפי (PyQt/Tkinter).
- לתרגל הצגת רשימה דינמית עם כפתורים עבור כל קובץ.

## מבנה קבצים:

```
/project  
  file_server.py  
  file_client.py  
  /server_files  
    ← תיוקינה שבה נשמרים הקבצים בשרת
```

## דרישות המערכת:

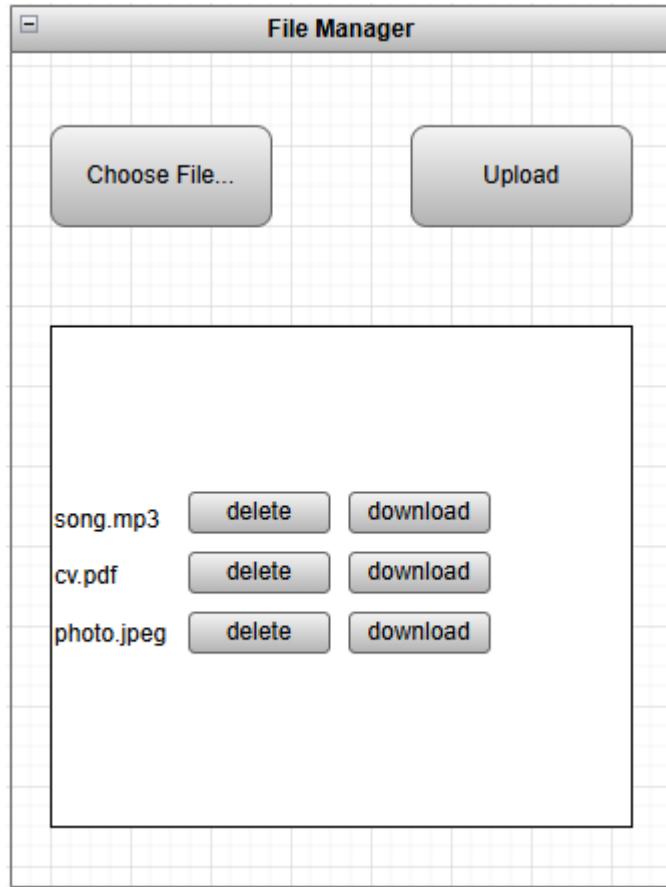
### צד שרת (file\_server.py)

- מקבל חיבורים מלוקחות.
- תומך בפקודות:
  - o upload – מקבל קובץ ושומר בשרת.
  - o list – מוחזר רשימת קבצים.
  - o download – שולח קובץ לפי שם.
  - o delete – מוחק קובץ לפי שם.

### צד לקוח (file\_client.py)

- ממשק גרפי פשוט עם:
  - o כפתור בחירת קובץ מתוך מערכת הפעלה.
  - o כפתור העלאה לשרת.
  - o איזור המציג את רשימת הקבצים מהשרת, כאשר ליד כל קובץ:
    - כפתור הרודה
    - כפתור מהיקה

## סקיצה של הממשק (PyQt או Tkinter)



## פרוטוקול תקשורת (הודעות פשוטות):

- כל הודעה בצורה JSON:

```

json
# בקשה העלאה
{ "command": "upload", "filename": "file.txt", "size": 1024 }

# בקשה רשימה
{ "command": "list" }

# בקשה הורדה
{ "command": "download", "filename": "file.txt" }

# בקשה מחיקה
{ "command": "delete", "filename": "file.txt" }

```

## תפקיד:

- לאחר העלאה – רשימת הקבצים מתעדכנת.
- הורדה שומרת קובץ ל-'Downloads' או ספירה לבחירה.
- מהיקה מוחקת רק מהשרת.
- העלה נבדקת לפי גודל סביר (לא < 16MB)

---

## הרחבות אפשריות:



- הציגת גודל ותאrik כל קובץ.
- תיירות.
- תמיינה בסיסמא/הרשאות.
- drag-and-drop עם GUI