

שם התרגיל:

שמירת קבצים על השרת עם ממשק גרפי ללקוח

מטרות:

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

מבנה קבצים:

```
project
├── file_server.py
├── file_client.py
└── server_files
    └── ← תיקייה שבה נשמרים הקבצים בשרת
```

דרישות המערכת:

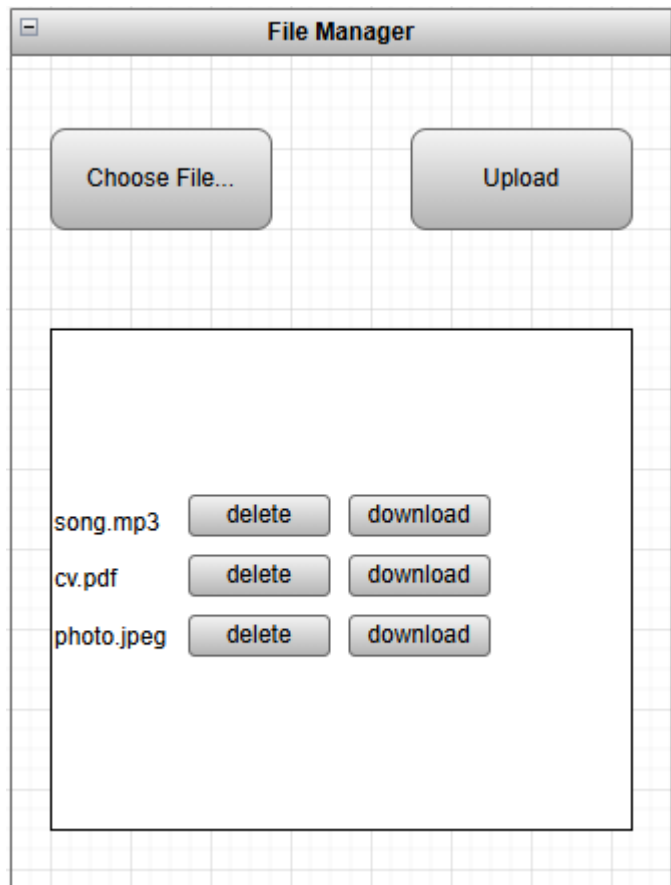
צד שרת (file_server.py)

- מקבל חיבורים מלקוחות.
- תומך בפקודות:
 - 0 upload – מקבל קובץ ושומר בשרת.
 - 0 list – מחזיר רשימת קבצים.
 - 0 download – שולח קובץ לפי שם.
 - 0 delete – מוחק קובץ לפי שם.

צד לקוח (file_client.py)

- ממשק גרפי פשוט עם:
 - 0 כפתור בחירת קובץ מתוך מערכת ההפעלה.
 - 0 כפתור העלאה לשרת.
 - 0 אזור המציג את רשימת הקבצים מהשרת, כאשר ליד כל קובץ:
 - כפתור הורדה
 - כפתור מחיקה

סקיצה של הממשק (PyQt או Tkinter)



פרוטוקול תקשורת (הודעות פשוטות):

- כל הודעה בצורת JSON:

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

# בקשת רשימה
{ "command": "list" }

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

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

תפקוד:

- לאחר העלאה – רשימת הקבצים מתעדכנת.
- הורדה שומרת קובץ ל-Downloads או ספרייה לבחירה.
- מחיקה מוחקת רק מהשרת.
- העלאה נבדקת לפי גודל סביר (לא $16\text{MB} <$).

הרחבות אפשריות:

- הצגת גודל ותאריך כל קובץ.
- תיקיות.
- תמיכה בסיסמא/הרשאות.
- GUI עם drag-and-drop.