

**Lebanese American University**  
Department of Computer Science and Mathematics

CSC 430 Computer Networks  
Spring 2024-2025

DESIGN PROJECT  
Advanced File Sharing System

## 1. Description

In this project, you will develop a multithreaded client-server file sharing system that will allow users to upload, download, and list files in a distributed environment with additional BONUS functionalities like a GUI/web interface, access control, progress tracking, and resumable downloads.

## 2. Core Requirements

Write a python code to implement the following functions in your proxy server:

### A. Client-Server Architecture

- A **server** hosts shared files and allows multiple clients to connect.
- Each **client** can upload, download, and list files.
- The system must support **multiple clients simultaneously** using **multithreading**.

### B. File Operations

- **Upload Files:** Clients can send files to the server for storage.
- **Download Files:** Clients can request files from the server.
- **List Available Files:** Clients can retrieve a list of files stored on the server.

### C. Network Communication (Sockets)

- Use **TCP sockets** to ensure reliable file transfers.
- Implement a **custom request-response protocol of your choice** instead of using HTTP/FTP.
- Example:
  - UPLOAD filename size → Client initiates file upload.
  - DOWNLOAD filename → Client requests a file.
  - LIST → Client requests the list of available files.

### D. File Integrity Checking

- Implement any error detection mechanism (example **MD5 or SHA-256 hashing or even checksum**) to verify file integrity after upload/download.
- Prevent file corruption during transmission by comparing hashes before and after transfer.

#### E. File Duplicates

If a file with the same name exists:

- Allow **overwriting**.
- Automatically **rename the new file** (for ex: filename\_v2.txt).
- Maintain **version history** of previous files.

#### F. Logging System

- **Server-side logging** to track client connections, file transfers, and errors.
- **Client-side logging** to track file transfers and errors.
- Store logs in a file with timestamps and event details.

### 3. Bonus Features

Following is a list of bonus features that you may want to add to your project for additional grades.

#### A. GUI or Web Interface

##### Option 1: GUI (example Tkinter or PyQt)

- A desktop-based graphical interface where users can:
  - Upload/download files with buttons
  - View file transfer progress bars
  - List available files in a structured format

##### Option 2: Web Interface (example Flask or Django)

- A web-based system accessible via a browser with features:
  - File uploads/downloads via a web dashboard
  - File list with details (size, timestamp)
  - Transfer progress bar using AJAX/WebSockets

#### B. Access Control

- Implement **user authentication** (username/password) before accessing files.
- **Role-based access:**
  - **Admin:** Can delete files and view logs.
  - **User:** Can only upload/download files.
- Store credentials securely in a **database (SQLite, MySQL)**.

#### C. Resume Interrupted Downloads (Checkpointing)

- If a download is interrupted due to network failure, allow resumption from the last received byte instead of restarting.
- Implement **file chunking** where files are sent in parts instead of all at once.
- Store metadata (file size, last received chunk) to track incomplete downloads.

#### D. Progress Bar for File Transfers

- Use a **GUI progress bar** to visually display file transfer progress.
- Update the progress bar dynamically as file chunks are sent/received.

#### 4. Grading

In this project you will work in the assigned groups and the grading will be based on the code, the report and the live demo.

#### 5. Deliverables

At the end of this project, you should submit your (thoroughly documented) code along with a project report. In this report, you should describe your high-level approach, the challenges you faced, a list of properties/features of your design, and an overview of how you tested your code. Your report should explain each of the required functions. You should also include screen shots of your running application.

- The submission of the Final Report with the full application and a readme file is due on **the beginning of the last week of the semester on Blackboard.**
- Project Demos will be held in the **Last week of the semester.**

#### **Important Notes:**

- The bonus parts may be worth up to an additional 20% on the total project grade.
- The grade includes the implementation decisions you have taken and the testing you have performed and how you explain these in your report.
- Each member of the group will be graded **individually** based on her/his input to the project.
- Code comments **MUST** clearly indicate which team member contributed which part of the code. Comments should also clarify the code.
- Code used from other sources (if any) must be clearly documented and flagged in the code by a comment.