Project-2

You need to create a file sharing protocol with functionalities like download and upload for files and indexed searching.
Following features need to be implemented :

- The system should have 2 clients (acting as servers simultaneously) listening to the communication channel for requests and waiting to share files (avoiding collisions) using an application layer protocol(like FTP/HTTP).
- Each client has the ability to do the following :
    - Know the files present on each others machines in the designated shared folders.
    - Download files from this shared folder
- File transfer should incorporate MD5 checksum to handle file transfer errors.

Specifications : The system should incorporate the following commands:

- IndexGet flag(args)
    - Can request the display of the shared files on the connected system
    - The history of requests made by either clients should be maintained at each of the clients respectively.
    - The flag variable can be shortlist , longlist .
    - **Shortlist (BONUS marks for implementing with particular extension)**:
        - Flag would mean the the client only wants to know the names of files between a specific set of timestamps The sample query is shown below:
            - $> IndexGet shortlist <starttimestamp> <endtimestamp>
            - Output : should include ‘name’ , ‘size’ , ‘timestamp’ and ‘type’ of the files between the start and end time stamps.
            - *****BONUS***** Return only *.txt , *.pdf files between specified time stamps with this type of

                query $> IndexGet shortlist <starttimestamp> <endtimestamp> *.txt or *.pdf

    - **Longlist :** flag would mean that client wants to know the entire listing of the shared folder/directory including ‘name’, ‘size’ , ‘timestamp’ and ‘type’ of the files .
        - $>IndexGet longlist
        - Output : similar to above , but with complete listing.
        - *** BONUS*** : Return longlist for only *.txt file contained a word “Programmer” in it.

- **FileHash flag(args) :**
    - This command indicates that the client wants to check if any of the files on the other end have been changed. The flag variable can take two values, verify and checkall.
        - **Verify :** flag should check for the specific file name provided as command line argument and return its ‘checksum’ and ‘last modified’ timestamp.
            - $>FileHash verify <filename>
            - **Output :** checksum and lastmodified timestamp of the input file.
        - **Checkall :** flag should check perform what ‘verify’ does for all the files in the shared folder. (Hint: this command can be used for the periodic check of changes in the files of shared folders)
            - $> FileHash checkall
            - **Output :** filename , checksum and last modified timestamp of all the files in the shared directory.
- **FileDownload flag(args) :** This would be used to download files from the shared folder of connected user to our shared folder.
    - The flag variable can take the value TCP or UDP depending on the users request.
    - If a socket is not available , it should be created and both clients must use this socket for the file transfer.
        - $>FileDownload <filename>
        - **Output :** should contain the filename , filesize , lastmodified timestamp and the MD5hash of the requested file.

Caching flag(args) : It(Client) checks if the requested object is cached (i.e. Client already has the request webpage or file), and if yes, it returns the object from the cache, without contacting the server.

The size of cache should be limited and once it crosses the limit the files which haven’t been used recently need to be removed and the new files should be inserted in the cache

- $>Cache verify <filename>
- **Output :** If the file is present return it from cache else “FileDownload” it and update the cache.
- $>Cache show
- **Output :** Should print all elements of the cache and their sizes.

Instructions :

- You are allowed to choose any language like C , C++ , Python.
- Plagiarism of any kind (online or your batchmate’s or past batches codes) will be given **zero marks**