Champlain College - Lennoxville
Lab 3: Storage server

PROGRAM:	420.B0 Computer Science Technology	
COURSE:	Transactional Web Applications 1	
COURSE CODE:	420-430-LE	
WEIGHT:	6% of the final score	
SEMESTER:	Winter 2023	
INSTRUCTOR:	Francis Gauthier fgauthier@crcmail.net	Office C-239

Context

A web server can be used as a storage solution. Most data hosted on the cloud are simply stored on large storage disk and accessed infrequently.

Your task is to build a web server that will be used as a storage solution. Your server must provide four actions possible:

- The ability to upload a file
- The ability to retrieve a file previously stored
- The ability to clear the storage
- The ability to monitor the storage limit

Using Express and Multer, as seen in class, you must build a viable storage solution with a web server.

Storage limit

A web server uses the disk space available by the machine host. While receiving several files, the server could bloat up the machine resources. To avoid that issue, we will set a hard limit on the amount of space dedicated to the storage server. To free up some space, we will also allow the user to remove some files or clear the storage completely.

Your server should only accept to store items using **200 MB of storage**. For example, if a file is 10MB large, but the current space used is already at 195 MB, then that file will be refused.

Storing a file

Your web server must provide an interface to upload a file and keep it stored on the disk on the server.

The server should provide some sort of **unique identifier** that allows to retrieve the file efficiently later.

Constraint: The upload must be made through an HTTP request with a multipart-form data body.

Constraint: If the file size is too large and would bust the storage limit, then the HTTP request must be refused with a 4XX status code.

Retrieving a file

Using a **unique file identifier**, a user should be able to retrieve a file from the web server at any time.

When a file is retrieved, it is not removed from the server storage. It can be accessed several times this way.

Constraint: The file retrieve must be made over an HTTP request.

Clearing the storage

There must be a way to clear up some space on the storage server. You must provide a way to clear up some space with two different methods (routes):

- By specifying the **unique file identifier**
    - Only one file is removed or none, if the identifier does not match
- By clearing up all the space at once

Constraint: The two ways of clearing up space must be made through HTTP requests, DELETE methods

Monitor the storage space

There must be a way to see how much storage space is used by the server. Through an HTTP request, send back the following information about the storage used:

- Number of files present in storage
- Maximum space allowed (200 MB)
- Current space allocated to files
- Percentage of space left to use

Evaluation

Criteria	**Weight**
Server	
Ability to upload a file	/1.00
Ability to retrieve a file	/1.00
Ability to clear the storage	/1.00
Ability to monitor the storage limit	/1.00
Documentation	
Documentation on how to install and start the server	/0.50
Use of HTTP requests is standard and easy to understand	/0.50
Documentation on how to use the different routes of the server	/1.00
Total	/6.00

Submission

Files to submit:

- A zipped folder containing:
  - Your web server
  - A documentation file (docx, pdf, markdown, etc.)
- Source code bigger than 10MB will not be accepted -> Delete your *node_modules* before zipping or ignore them when zipping.

Submission must be made through LEA. Submission deadline is Friday April 14th End Of Day. Late submission refused.

The students can submit up to 3 times and will receive their grade and feedback each Friday following their submit.