Assignment 3: Completing the CRUD Operations for the To Do Server

Context/Scenario

This assignment uses the code developed in the lecture for the to do list. So far, all CRUD operations have been implemented except for the update operation. In this assignment, you will be demonstrating the implementation of the update operation for the To Do List Server API

Instructions:

The assignment consists of two parts.

Part 1 – Base Code: The task for this part is completed offline and represents the base of your code that you will add to in Part 2. The code for Part 1 must be submitted as a separate file.

Part 2– Skill Demonstration: The task for this part will be recorded **as completed**. Note that in your recording it is not sufficient to only write the code, but you must also explain the logic and describe how your logic is being implemented. In addition, your recording must show a sample run that demonstrates the correct execution of your code.

Important Guidelines:

- The recording **should not exceed 15 minutes** in duration. Otherwise, the recording will not be considered.
- The code for Part 1 should be the starting point for your implementation of Part 2.
- The code for Part 2 must be developed during the recording. Ready-made code will not be accepted.
- To ensure that your recording will be accepted for grading, it is essential to show both your face and your screen while demonstrating the development steps of the code and explain each step as it is developed.
- Use Kaltura software for recording your video. Kaltura is freely offered by GMU. More information can be found here
- Your presentation needs to be clear, concise, and effectively communicate the steps and logic of your solution.
- Use the grading rubrics as a guide to prepare your presentation.
- Using Frameworks and/or advanced techniques not covered in class will result in a 0 grade.
- Providing code that is copied from ChatGPT and reading the accompanying explanation generated by ChatGPT will not be accepted as a valid skill demonstration for this assignment. This practice will result in a grade of 0.
- Resubmission of the assignment after the due date is not allowed.

Please ensure that your recording adheres to these guidelines to receive full credit for your submission.

Assignment Description

Part 1 – Base Code: Reading the Server Port from the Command line

The to-do list server developed in the lab has a hard coded port value 8080. In this part of the assignment, you need to update the code such that the port number is passed as an argument from the command line. In case the port number is missing, or the port number is below 3000 the server script will issue an error message and terminate. Otherwise, the script will start the server and print out the message 'The server running on port **PORT#**' where **PORT#** is the number of the port passed as a command line argument
Check the Sample Run for the server-side below

Sample Run:

```
% node todoServer.js
Missing Server Port Number
Usage: node todoServer.js [port number]

% node todoServer.js 2000
Port number must be greater or equal to 3000

% node todoServer.js 8000
Server running on port 8000
```

Part 2– Skill Demonstration: Completing the Code for the PUT Method

In this part of the code, you will complete the implementation of the PUT method. The PUT method will allow the client to update one of the entries in the To-Do List. The PUT method implementation can be achieved by combining the code from the POST method and the DELETE method. To update an item, the server first needs to identify the old item using the index value sent by the client in the curl request similar to the DELETE method, and then reads the updated text and adds it to the list similar to the POST method.
Check a sample run for the PUT method below. The server responses are colored green.

Sample Run for PUT method

```
hhassa2@L-MAC-N04091 Chapter4 % curl localhost:8080
1)go to the dentist
2)buy the books
3)Complete lab 2%
hhassa2@L-MAC-N04091 Chapter4 % curl -X PUT  -d 'Complete Assignment' localhost:8080/2
OK
hhassa2@L-MAC-N04091 Chapter4 % curl localhost:8080
1)go to the dentist
2)Complete Assignment
3)Complete lab 2%
```

Deliverables

JS Files: Submit all your .js files as one zipped folder. Your folder will include the following:

- A .js file for part 1
- A .js file for part 2

Video Recording: Submit your video recording using the 'Create Submission' option and upload your video from Kaltura. More information can be found here.

Grading Rubrics:

This assignment is graded out of **70**

Part 1: 10 Points

Criteria	Points
Comment describing the task for Part 2 at the beginning of the file	**5 points**
The code submitted for Part 1 coincides with the starting code shown in the recording for Part 2	**5 points**
Total Part 1	**10 points**

Part 2: 60 Points

Criteria	Points
Comment describing the task for Part 2 at the beginning of the file	**5 points**
Overview of Part 1 Code	**10 points**
• Description for the code that checks the correct invocation of the server.	5 points
• Description for the code that checks the value for the port	3 points
• Description for the code that prints the port value to the screen	2 points
Identifying the item at index similar to DELETE Method	**30 points**
• Implement and explain the logic for extracting the index from the req.url	5 points
• Implement and explain the logic for checking on the validity of the index value	5 points
• Implement and explain the logic for responding to client in case of an invalid index value	5 points
• Implement and explain the logic for handling a two-digit index	5 points
• Implement and explain the logic for item at index is not found	5 points
• Implement and explain the logic for responding to client in case of item at index is not found	5 points
Reading text like the POST Method and updating list	**15 points**
• Implement and explain the logic for reading the incoming item	5 points

• Implement and explain the logic for updating the item at the specified index	5 points
• Implement and explain the logic for responding to client in case of successful update	5 points
Demonstrating the Sample Run	10 points
Total Part 2	**70 points**