|  |  |
| --- | --- |
| Instructions |  |
| **[Please post the urls of part B and ( C if you do bonus) of your hosted assignment with  https/http at the comments section before hitting submit ]**   * **Note! you cannot use nodejs express for this lab**   **A(2 mark) debugging.jpg**  **In Visual studio code create a module, math.js, with two functions to add and subtract two numbers.**  **call the math module functions in your code, app.js, and place a breakpoint ( to practice debugging)**  If you dont know how to install nodejs, follow instructions attached to this lab: [Install-debug-nodeJs-Visual-Studio-Code.pdf](https://learn.bcit.ca/d2l/common/viewFile.d2lfile/Database/ODMzNDQxMg/Install-debug-nodeJs-Visual-Studio-Code.pdf?ou=782287) (attached) to set up your local machine for debugging   you need to prove that debugging is working in your machine  Place a breakpoint in your code, with console.log statement like the one in the image below  **note:**  1- Use  backtick format  2- replace YOURNAME with your first name ( so that you don't accidently submit somebody else's screen shot)  start debugging and make screenshot proving that the breakpoint pauses execution of your script, just like in the pdf file)  for part A you only need to submit the screenshot to prove that the debugging works and the breakpoint hits! e.g.  Text  Description automatically generated  **B (8 marks) simple API call to get the time of server**  In part B you implement your very first API server. So far, you have been a consumer of third party APIs ( such as google APIs, Weather API etc), now you are going to take your very first step to make your own.  Develop a server side nodejs app to get your name you sent via browser address bar,  greet you and return the current time of the server ( entire message in blue)\*. You need to host it remotely\*\*  \* the message the server returns has to contain the in-line styling to be displayed in blue. Don't style the message in the client side.  \*\*again as stated in lecture notes and the learning hub, it is your responsibility to obtain a web hosting service to host your assignment.  In your browser you want to send your name to the server from the address bar of the browser.  **Example**:  an API request like this one at the browser's address bar  [https://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/getDate/?name=John  Shall return this message in **blue**:  **Hello John, What a beautiful day. Server current date and time is \* (e.g. Wed Sept 01 2021 12:52:14 GMT-0800 (Pacific Standard Time)**  \*: whatever current date and time of the **server**is. So it has to come from the server, not from your browser.  Your part B project must be composed of two files:  1- the **server.js**  2- .**/modules/utils.js** for the definition of your function getDate() {...  ( you need to put your function in a module)  Note: As you can see, there is no need to have any client side implementation  **Bonus: ( please mention Bonus at comment section)**  **C) ( 1 marks ) Writing to a file, Reading from a file**  **C.1**  [https://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/**writeFile**/?text=BCIT  which appends a new line with text BCIT to the text file file.txt  You need to append the text (which includes in the url) to the exiting file (file.txt ). Do not re-create the file every time the user sends the request.  Append means: if the file does not exist, create it and then store the text to it if the file exists, just add ( append) the text of query string to the bottom of the existing file.  **C.2**  [https://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/**readFile**/file.txt  which reads the entire content of the updated file file.txt and returns it back to the browser and displays it at the browser's page  if the file does not exist, your app has to return 404 error message including the file name the user had entered  **Deliverable:**   * Upload the **screenshot of part A** and the urls of part B ( 1 url) and, if you are doing the bonus part C ( 2 urls) , e.g. * together with your urls of part B ( 1 url ) and ( if doing part c, **2** urls) with https/http at comment section * e.g. * **B**- [**https**://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/getDate/?name=John ( ***replace John with your name***) * if doing the ***bonus***  ( no bonus mark will be received if urls are not posted) ( please mention "Bonus:" at the comment section so the marker knows you did the bonus) * **C.1**- [**https**://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/writeFile/?text=BCIT * **C.2**- [**https**://yourDomainName](https://johngates.ca/comp4711/labs/1/quizCreator/).ca/COMP4537/labs/3/readFile/file.txt * At the time of marking, we try them with various the url query strings ( we change John and BCIT to see the result accordingly)   **Rubrics (additional d**eductions )   * (-6) if Part B is not hosted               (-6) if the url(s) are not posted at the comment sections  ( which is as equal as not hosting at all)              (-2) if you post the urls at the comment section but forgot to include http:// or https://  ( -4 ) if making the message in blue color is not done or is on the client side  Bonus will be marked, only if parts A and B are done and complete  -10%  mark deduction for each day late, 0 after three days late submission | |