Implement a minimal dictionary REST API using Node.js with the ability to store and search for a definition of an English word.
For this exercise, the definitions do not have to be precise.

Service root endpoint: .../api/definitions/*

.../api/definitions/?word=book (GET request to get the definition of a word)
.../api/definitions (POST to create a new entry e.g. you put the query string in the body of your request (in the body not in the url, unlike GET) written or printed work consisting of pages glued or sewn together

In this lab you need to use two different web hosting accounts hosted on two different machines (Feb14: to clarify, two different hosting provider. Both cannot be on Vercel). One part of your assignment will be deployed(hosted) to one hosting service, the other part will be hosted on another hosting service. The two will communicate via API calls.

* this is the suggesting endpoint. you can pick the uri endpoint that works better for you

Server 1
The client side files (HTML or the js which initiate the AJAX calls) are being hosted at server1 (e.g. https://yourDomainName1.xyz*). It shall send AJAX API calls to the API service which is hosted at another server, server 2.

Server 2
The server2 (https://yourDomainName2.wyz*) that hosts the node js app that receives the API request from server1.
If it is a POST request, it adds a new record (word: definition) to our array of objects** (array of word:definition). If the word already exists, it returns a message. (e.g. 'Warning! blah blah ' already exists. Otherwise it adds the word and its definition to your dictionary and returns another appropriate message such as:

Request # 102 (the total number of requests the server served so far) (**updated on Feb14**: its also nice to return the updated total number of entries exist. This means the total number of words that exist after this recent update.)

New entry recorded:

"Book : A written or printed work consisting of pages glued or sewn together"

*: http or https (but if you are sharing the https, make sure ssl is already installed and your web pages do not issues any security risk warning)

**: please pick dictionary as the identifier of the variable that holds the array of entries (pick 'dictionary' as variable name for array of word:definition so that the marker can easily understand your code)

Required functionality

1. A POST request for creating and storing the definition (creating new entries to our dictionary)
2. A GET request for retrieving a definition given an English word.

The service should return the response as JSON

Store the data in memory (**note: we are talking about your server side memory, nodejs script**) , using array or any data structure you deem appropriate, therefore you do not need to set up a database at this stage for this assignment.

Front End (Server1)

1. The **store.html** provides a **text box for the word to input and and an area box for definition and a submit button** for creating the definitions. When the user presses submit button, display feedback for success or failure

.../COMP4537/labs/4/**store.html**

2. A **search.html** page with a test box **search input** and a display area for the response.
When the search term is present in the dictionary, display the **term and its definition**;
Otherwise, indicate this to the user with an appropriate message (e.g. Request# 103, word 'book' not found!).

you can use the client's native XMLHttp API we covered in this course (or Fetch which is part of javascript) to process the requests.

Please implement simple input validation on the front and back end to accept only non-empty strings (disallow numbers, etc)

Back End (Server2)

app.js which serves the GET and POST requests as described above **and returns the number of requests the server received so far**.

Learning outcomes

You will learn how to implement a basic RESTful service and how to consume the service using the browser's native web API.

Technology stack

- Vanilla JS (ES6), front end.
  node js http and other built-in modules for the backend (you cannot use express or anything that need npm install)

Deliverables

1. Zip everything in **YourTeam#Lab4.zip** and upload
At the learning hub comment section:
2. **Post URLs of the hosted application at the comment section** of learning hub
 2.1 the url(s) of the page for entry or retrieve on the server1 (or you can use same page for both entry and retrieval)
2.2 the url of the API server hosted on the server2

3. include Attribution to chatGPT if you used* (or any other resources, provided their license permits you) both in learning hub comment section and in your code)

Rubrics

8/10 for working and bug free application

- Working implementation of the RESTful service
  (3) Processing the POST request (updating the dictionary data in memory correctly and return the message if the entry already exists)
  (3) Processing the GET requests (look-up of the search term in the dictionary data)
  (1) Proper error handling, including sending appropriate status codes in the header of your response (https://www.restapitutorial.com/httpstatuscodes.html)
  (1) Proper packaging of response in JSON, as part of the json return the number of requests the server served so far

1/10 Working home page (with form for creating definitions)
1/10 Working search page (search widget and output area for displaying definitions)

Deduction rubrics

-3 mark will be deducted if URLs are not posted correctly and hyperlinked at comment section of your learning hub submission) (you need to include https and press space bar when you post your url)

(-4) if the part client and server are from the same origin (e.g. if are hosted on the same server) (if you don't have a teammate, please find a free service to host the client side (server1) as it is all static files and you don't need a DB or backend scripting; remember it has to have https)
(-4) if you use JQuery or other libraries/modules not allowed in your program

10% mark deduction for each day late submission (0 after three days late submission)

-2 if you use anything except const, let for variable declaration

-2 if you do not store user facing strings in a separate file

-2 Each incorrect answer results in a 2-mark deduction for questions about your project code during in-person assignment marking. Familiarity with both your and your partner's code is required.

-4 if you are not accompanying your partner in-person while your lab being marked

Needless to mention, any other deduction will be decided while marking

***chatGPT**

as already mentioned in the course outline, using chatGPT is fine but you need to
1- mention that in BOTH your code comment and the learning hub comment . e.g.
ChatGPT-3.5 (https://chat.openai.com/) was used to code solutions presented in this assignment. otherwise would be treated as Academic Dishonesty
2- you have to be familiar with every single line of the code)
3- you take full responsibility of the work you submitted