INF653 Back End Web Development - Midterm Project Requirements

1) You will build a PHP OOP REST API for quotations - both famous quotes and user submissions

2) ALL quotes are required to have ALL 3 of the following:
   a) Quote (the quotation itself)
   b) Author
   c) Category

3) Create a database named "quotesdb" with 3 tables and these specific column names:
   a) quotes (id, quote, authorId, categoryId) - the last two are foreign keys
   b) authors (id, author)
   c) categories (id, category)

4) Response requirements:
   a) All requests should provide a JSON data response.
   b) All requests for quotes should return the id, quote, authorId, and categoryId fields.
   c) All requests for authors should return the id and author fields.
   d) All requests for categories should return the id and category fields.
   e) Appropriate not found and missing parameter messages as indicated below.

5) Your root project URL should follow this pattern ending in /api:
   https:/your-project-name.herokuapp.com/api

   The request URLs below should be appended to the root URL.

6) Your REST API will provide responses to the following **GET** requests:

Request:	**Response:**
/quotes/	All quotes are returned
/quotes/?id=4	The specific quote
/quotes/?authorId=10	All quotes from authorId=10
/quotes/?categoryId=8	All quotes in categoryId=8
/quotes/?authorId=3&categoryId=4	All quotes from authorId=3 that are in categoryId=4
If no quotes found for routes above	{ message: 'No Quotes Found' }
/authors/	All authors with their id
/authors/?id=5	The specific author with their id
If no authors found for routes above	{ message: 'authorId Not Found' }
/categories/	All categories with their ids (id, category)
/categories/?id=7	The specific category with its id

If no categories found for routes above	{ message: ‘categoryId Not Found’ }

NOTE: In the above examples, the parameter numbers are examples. You could change the authorId=2 or categoryId=5, etc. and the requests should still have the appropriate response.

7) Your REST API will provide responses to the following **POST** requests:

Request:	**Response (fields):**
/quotes/	created quote (id, quote, authorId, categoryId)

Note: To create a quote, the POST submission MUST contain the quote, authorId, and categoryId.

/authors/	created author (id, author)

Note: To create an author, the POST submission MUST contain the author.

/categories/	created category (id, category)

Note: To create a category, the POST submission MUST contain the category.

authorId does not exist	{ message: ‘authorId Not Found’ }
categoryId does not exist	{ message: ‘categoryId Not Found’ }
If missing any parameters	{ message: ‘Missing Required Parameters’ }

8) Your REST API will provide responses to the following **PUT** requests:

Request:	**Response (fields):**
/quotes/	updated quote (id, quote, authorId, categoryId)

Note: To update a quote, the PUT submission MUST contain the id, quote, authorId, and categoryId.

/authors/	updated author (id, author)

Note: To create an author, the PUT submission MUST contain the id and author.

/categories/	updated category (id, category)

Note: To create a category, the PUT submission MUST contain the id and category.

authorId does not exist	{ message: ‘authorId Not Found’ }
categoryId does not exist	{ message: ‘categoryId Not Found’ }
If no quotes found to update	{ message: ‘No Quotes Found’ }
If missing parameters (except id)	{ message: ‘Missing Required Parameters’ }

9) Your REST API will provide responses to the following **DELETE** requests:

Request:	**Response (fields):**
/api/quotes/	id of deleted quote
/api/authors/	id of deleted author

/api/categories/	id of deleted category
If no quotes found to delete	{ message: 'No Quotes Found' }

Note: All delete requests require the id to be submitted.

10) Your project should have a GitHub repository with a pipeline connected to Heroku. The project should utilize JawsDB on Heroku for the database. It is suggested to develop the database locally on MySQL first.

11) You will need to populate your own quotes database. You may choose to populate the database manually or with Postman to start out. A good site to find quotes by category (topic) is: https://www.brainyquote.com/ Minimum 5 categories. Minimum 5 authors. Minimum 25 quotes total for initial data. You will need these minimums to pass the tests for the project.

12) **Submit the following:**
    a) A link to your GitHub code repository (no code updates after the due date accepted)
    b) A link to your deployed project on Heroku
    c) A one page PDF document discussing what challenges you faced while building your project.

✅ Test your requests and responses with Postman: https://www.postman.com/downloads/

🚀 **For students who want an extra challenge (not required):**
Allow a "random=true" parameter to be sent via GET request so the response received does not always contain the same quote. The response should contain a random quote that still adheres to the other specified parameters. For example, this will allow users of the API to retrieve a single random quote, a single random quote from Bill Gates (author), or a single random quote about life (category).

Examples of Extra Challenge Requests and Responses:

Request:	**Response (fields):**
/quotes/?random=true	1 random quote
/quotes/?authorId=7&random=true	1 random quote from the specified author
/quotes/?categoryId=10&random=true	1 random quote from the specified category
/quotes/?authorId=7&categoryId=10&random=true	1 random quote from the specified author & category