Web Science

Quiz 1: February 21, 2025

Enter your answers directly into this document (with the exception of #2 and #3).
All answers should be In Your Own Words, using complete sentences with proper spelling and grammar.

Save this document as either a Word docx or a PDF. For all questions other than #2 and #3, you will not receive any credit for answers not placed in this document.

When finished with the quiz, put everything you wrote (this document, all code, etc.) in your personal GitHub repo in a folder named **quiz1**. Local development only for this quiz, but everything you want graded must be on GitHub in a directory name **quiz1** no exceptions.

1. **Short answers** (25 points): (Answer in complete sentences unless otherwise specified, explain your answers)

    a. (8) Fill out this table as to what should happen for each verb at each endpoint **according to the letter of the law.** (No need for complete sentences.)

Resource	**POST**	**GET**	**PUT**	**DELETE**
/news	append a new news article to end of existing DB of news articles	Retrieve list (potentially paginated) of all news ids	Bulk update attributes of every news article in database	Delete all of the news articles
/news/123	Create/add new attributes to news article with id '123' that were not there before	retrieve information regarding news article '123'	Update previously existing attributes of news article '123'	Delete news article '123' from the DB

    b. (7) What is a package.json file? What is it used for? How is it created?

package.json contains project properties as well as a minimum version of dependencies for a Node.js project. It contains all of the metadata for the Node.js project. It is created by running 'npm init' in your project directory. The package.json file is used to handle project metadata, along with other configurations. It is also responsible for the management of dependencies and scripts within the project.

c. (10) Describe **in exacting detail** the sequence(s) of transaction(s) for a frontend to request data from some external entity via Node/Express.

For this example, lets say the front end is trying to retrieve news articles using the same conventions as Quesiton 1a. First, the front end would need to have some component or element set up to actually present the information retrieved from the external entity. For this example, we can use a span element with a specfific id of 'article'. Then the frontend would need some sort of JS function called on page load that would make a fetch call to the external Node/Express entity. It is important that this call uses an endpoint that is actually handled by the entity. If so, the entity will catch this request and the app function corresponding to that endpoint will be activated. Thanks to express, the request will be stored in the 'req' object. The app function will then handle the URL and it's parameters accordingly, curating some sort of response, in this case a news article that we will say is represented by a JSON object. This response will be stored in the 'res' object also thanks to express and sent back to where the fetch request came from (the front-ends js function). Now that the function has the news article json response from the Node/Express entity, it can use Jquery or getelementbyid to locate the span element on the actual front-end and update it's inner HTML with the response it just got from the external entity. Thus presenting data retrieved from the external node/express entity to the user.

2. **Coding question**: (60 points) Here is a free API that does not require any API keys: https://frankfurter.dev/

   Create a new GET endpoint in Node/Express that converts currency.

   Creativity matters; try your best to integrate this new information into your frontend. Try as hard as you can to make it feel like it is a meaningful, conscious, intentioned feature of your app. How you do that is up to you. Go beyond the minimum (but remember that creativity doesn't have to be visual). If you need to, write a short README.md file that tells me what I should consider for creativity. (creativity: 20 points of the 60 available for this question)

   You may use any and all open source libraries you want for this coding question, so long as you cite them in a README.md file.

20 points for frontend (could be React, doesn't need to be React)
20 points for backend (extending your API to include this new info, serving it to your frontend)
20 points for creativity

3. (15) Ensure the package.json file for Q2 has no errors when I run npm install.

4. **Extra credit (+5)**: Before moving into its current building, which building on campus was the home of the Rensselaer Student Union?

The student club house was the previous home of the Rensselaer Student Union.