Web Science

Quiz 1: February 27, 2024

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 your **quiz1** folder.

1. **Short answers** (25 points): (Answer in complete sentences unless otherwise specified, explain your answers)
   1. (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** | Add news on API | Get /news Json or information | Change something in news | Delete news |
| **/news/123** | Add something like Json file on /news/123 | Get information from /news/123 | Change something in /news/123 | Delete some or all of thing in /news/123 |

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

A Json file which records metadata, this file is used for that the project always has the latest information about the libraries and tools needed to work correctly.

Created: npm -init -y

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

Set up the project structure with HTML and JavaScript files.

Create an HTML form in the frontend to capture user input.

Use JavaScript to send a POST request to the backend using the Fetch API.

Create a Node.js server using the Express framework.

Process the form data on the backend and return a JSON response.

1. **Coding question**: (60 points) Here is a free API that does not require any API keys: <https://www.boredapi.com/> – the documentation for which can be found here: <https://www.boredapi.com/documentation>  
     
   Create a new GET endpoint in Node/Express that accepts the type of activity (and optionally additional information in a query string–check the documentation!) from your frontend and returns the API’s information about that activity back to the frontend. If the user fails to input a valid activity, return the /api/activity endpoint (which returns a random activity).  
     
   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. Don’t make it look like some random afterthought. 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

1. (15) Ensure the package.json file for Q2 has no errors when I run npm install.
2. **Extra credit (+5)**: Before moving into its current building, which building on campus was the home of the Rensselaer Student Union?

87 gym?