Web Science

Quiz 1: March 1, 2022

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: answers.docx (or .odf or .pdf) (-5 if wrong name). 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.) on GitHub into a branch in your lab repo named: quiz1 (-5 if submitted incorrectly). **Do not submit your node\_modules folder! (-15 if you submitted the node\_modules folder)**

1. **Short answers** (25 points): (Answer in complete sentences, explain your answers)
   1. (5) How can I determine the type of device that my page is being displayed on? Give two examples of why I might care.
      * 1. NPM library provides many good methods to detect the device type and “react-device-detect” is one of them. It tells whether the device is either Mobile or Desktop.
        2. To use old UserAgent String with a custom-defined match is also one of the ways to determine the device type. The technique is less recommended than the first one as userAgent might get changed in future versions and any custom code written manually will no longer be used.
   2. (5) What is a package-lock.json file? What is it used for? Is it required?

Package-lock.json is automatically generated file for any operations where npm modifies either the node\_modules or package.json. Package-lock.json is the file that keeps track of the exact tree generated for every package installed so that a product is reproducible in the same way even if packages are updated by their maintainers. While it is recommended to have both package-json and package-lock.json, since package. Json is the one required for a project as it records the minimum version needed for an application, it is not mandatory to have package-lock.json file.

* 1. (5) What is npm? How does it work? Why is it used?

Npm is the package manager for the Node JS. It puts modules in place so that node can find them and manages dependency conflicts efficiently. Having a node project with a package.json file, “npm install” would install all the dependencies listed in the package json file. Npm is broadly used because it makes developers easily share and update code using the npm’s central registry.

* 1. (10) Describe **in detail** the sequence(s) of transaction(s) for a frontend to request data from some external entity via Node.
     1. 1) Frontend requests an access to an endpoint in REST API.
     2. 2) Node calls an API and directs to the specified endpoint.
     3. 3) then upon its type of request (get/post/put/delete), it follows its login under its api call.

1. **Coding question**: (40 points) Create a webserver in node.js, name your server: server.js. You may use Express, but you *may not use a generator* – (i.e., NOT express-generator), which will serve a simple frontend (in the technologies of your choosing). The frontend will provide an input field for ZIP code and a series of buttons that issue GET and/or POST requests when clicked to the Node server. (frontend: 10 points)  
     
   Upon entering a ZIP code and clicking the “Temperature” button, your application should send a POST request to <http://localhost:3000/temperature>. Node should then get the current temperature for that ZIP code (I bet you have an API for that!) and send the frontend back that information. The frontend should then output a sentence that says the name of the location and whether it is Freezing (<33F), Cold (between 33 and 50), Warm (between 51 and 80) or Hot (>80) – display the corresponding message in a unique color for each category. (temperature sequence: 10 points)  
     
   Upon clicking the “Is RPI windy?” button, your application should send a GET request to <http://localhost:3000/wind>. Node should get wind speed information for Troy, NY, via that API and send that information back to the frontend. Have the frontend display this information in a unique color. (wind sequence: 10 points)  
     
   Creativity matters; don’t just give me an empty white page with a text entry form box and two buttons. Go beyond the minimum (but remember that creativity doesn’t have to be visual). If you need to, write a short README file that tells me what I should consider for creativity. (creativity: 10 points)  
     
   ***You may use any and all libraries you want for this coding question.***

1. (15) Ensure the package.json file for Q2 has no errors when I run npm install & run your code.
2. (20) Provide **two** different explanations of the code below. The first explanation should be a high-level explanation (no less than four complete sentences) outlining what this code does to someone who has no coding experience. The second explanation should be a *detailed* one explaining line-by-line what the code does. If there are any errors in the code, fix them.

var net = require('net');

var sockets==[];

var s = net.Server(function(socket) {

sockets.push(socket);

socket.on('data', function(d) {

for(var i=0; i<sockets.length;i++) {

if (sockets[i]==socket) continue;

sockets[i].write(d);

}

});

socket.on('end', function() {

var i=sockets.indexOf(socket);

sockets.splice(i,1);

});

});

s.listen(8080);

Answer:

1) The code above is the use of socket.io.

It first imports a required api to use a socket.

Then it makes an empty space to append socket.

While the server is on, it receives a data and push them into created space.

Then it goes through each element in the space and update it's value unless they are same.

Finally, when the system hits 'end', it updates an index of socket and splice the space.

2) It firsts imporst the net module which provides an asynchronous network API and store it to variable 'net'.

Then it creates an empty array named 'sockets' and push socket parameter passed in to the array using function.

Then the socket server receives 'data' and run a foor loop for the length of sockets array and unless i-th element equals to socket data, it writes 'd' to the i-the index of slot in the 'sockets' array.

If socket reaches 'end' it sets variable i to the index of socket in the array then splice the array using 'i'.

The server listens at localhost:8080.

1. (+5) What is the name of the RPI-developed chat protocol popular in the 1990s?
   1. Submitty