# 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)

    a. (5) How can I determine the type of device that my page is being displayed on? Give two examples of why I might care.

    We can detect the user-agent `Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36` which can show what device is the user using

    Another way to detect we can detect the window size by detecting window size we can possible to speculate the device

    b. (5) What is a package-lock.json file? What is it used for? Is it required?

    The package-lock.json is a file generated during npm install to record the specific source and version number of each npm package actually installed in the current state. Because npm is a manager for managing dependencies between packages, it allows developers to mark the dependencies of their projects on npm libraries in pacakge.json. You can choose to indicate the version of the library package you need in the following ways

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

    NPM is a package management tool installed with NodeJS. It can solve many problems in NodeJS code deployment.

    - Allows users to download third-party packages written by others from the NPM server for local use.

- Allows users to download and install command-line programs written by others from the NPM server for local use.
- Allows users to upload their own packages or command-line programs to the NPM server for others to use.

d. (10) Describe **in detail** the sequence(s) of transaction(s) for a frontend to request data from some external entity via Node.
There are two ways that the frontend can get data from the backend by using sock.io and API. socket.io is a module of Node.js. It provides a simple way to communicate through WebSocket. The WebSocket protocol is very complicated, but Socket.IO provides a server and a client. components on both sides, so only one module is required to add WebSocket support to an application. And also supports different browsers. By using API, the frontend can either use GET/POST/PUT/DELETE, it will send to the backend and find the corresponding verb and endpoint to do the action.

2. **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.***

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

4. (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'); //creating both servers and clients

var sockets==[]; //create a sockets array

var s = net.Server(function(socket) {
   sockets.push(socket); //push the socket into sockets array

   socket.on('data', function(d) {
      for(var i=0; i<sockets.length;i++) { // loop through all
the sockets
         if (sockets[i]==socket) continue; // if there is data
ignore it.
         sockets[i].write(d); // if not write the data in
sockets[i]
      }
   });
   socket.on('end', function() {
      var i=sockets.indexOf(socket); //check the index
      sockets.splice(i,1);  // add it into corresponding
location of the array
   });
});

s.listen(8080); //listen at port 8080
```

This program creates a client that listens to the port 8080, later on it creates an array of sockets, it contains the data in the socket(sockets.push(socket);). Additionally it use socket.on to listen to incoming data. It loop through the sockets to check if there have same data in sockets compare to socket information in socket if not the item in the array will write the data d inside. The socket.on("end") is getting the index of the socket item and add in to the array.

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