Web Science

Quiz 1: March 10, 2016

100 points max

Place your name on the top of the document in the header

Enter your answers directly into this document (with the exception of #2 and #3)

All answers should be in be in Your Own Words, and use proper grammar

Make sure your answers use an alternative font and/or color

Save the document as

ITWS4500-S16-Quiz1-*yourRCSID*.docx

Place all documents/files including this one in a folder named

ITWS4500-S16-Quiz1-*yourname*-*yourRCSID*

When finished with the quiz, zip your folder and all related files into a file named

ITWS4500-S16-Quiz1-*yourname*-*yourRCSID*.zip

And submit it to LMS

1. **Frameworks** (25 points): (Answer in complete sentences, explain your answers)
   1. (5) What is a Media Query? How is it used? Why do we use them?

**A media query is a feature that was introduced in CSS3, though previously in CSS2 it was called media types. It allows us to display content differently, depending on the screen size or resolution. Media queries are used in order to provide the best experience for each type of user. Specifically, media queries look at the device and can be used to check the following: resolution, orientation of device, width and height of viewing window, and width and height of device. Within CSS you must specify your media query and all of the CSS that depends on the query.**

* 1. (5) What is Bootstrap? How is it used? Why is it used?

**Bootstrap is a front-end (HTML, CSS, and JavaScript) framework that you can use for a basis for creating websites or web applications. Included in Bootstrap is HTML and CSS based design templates that are all responsive. Bootstrap was created with the idea of “mobile first” (designs need to work across all platforms). Additionally, it has optional JavaScript extensions. One of the features of Bootstrap is that you can use it as a base for the CSS on your website and then customize by creating a style.css file. A few reasons we use Bootstrap are the speed of which you can develop a website, all templates are responsive, it is consistent, you can customize it, and there is a large support community if you run into any issues.**

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

**AngularJS is a client-side framework that can be used for dynamic web applications. Angular allows you to use HTML as a template and extend it’s syntax to express your apps components clearly and concisely. Furthermore, AngularJS was created to address many challenges faced when developing single-page applications. There are a few reasons AngularJS is used: it is a declarative user interface through HTML, all of its data models are JavaScript objects, code is more concise, DOM manipulations are in directives not the view, and it is unit testing ready.**

* 1. (10) Describe the difference between JavaScript and CSS frameworks. Provide at least 2 examples for each in your answer.

**JavaScript is an object-ordient computer programming languages that is used when creating interactive effects within browsers.**

**CSS (or front-end) frameworks usually consists of a package made up of files of standardized code in HTML, CSS and JavaScript documents. Typically the components are CSS source code to agreat a gride, typography style defniitions for HTML elements, standard CSS classes, and solutiosn for browser incompatibility. A few examples of CSS frameworks are Bootstrap, Foundation 3, Skeleton, YAML 4, Tuktuk, Kube, Groundwork and Gumby.**

**The difference between JavaScript and front-end frameworks is that JavaScript limits you as well as it is not as versatile as a framework. Oneo f the main limitations of plain JavaScript is that it depends a lot on the browser to run; each browser interprets JavaScript differently. A perk of using a CSS framework is that it has cross-browser compatability with very minimal differences.**

1. **Node.js** : (40 points) Create a webserver in node.js, using express – (NOT express-generator), which will output a simple HTML page with a button labeled ‘Run’ when GET request is received on <http://localhost:3000>. Upon clicking the button, the server should get the current temperature in Spokane, WA and output a sentence that says whether it is Freezing (<10F), Cold (btw 10 and 40), Warm (btw 40 and 70) or Hot (>70) – display the corresponding message in a unique color for each category.

**I was unable to finish this question. Currently, I have an index.html page that displays a button with the word run on it. When the button is pressed, it links to the function getWeather(). This function is broken in the sense that it doesn’t connect with the node server to get the temperature. Because I was unable to connect with the node server, I decided to console.log the sentences that corresponded with the temperature. When you run server.js it pulls up the current temperature for Spokane, WA. Then, depending on the temperature, it displays a sentence about the weather. Due to lack of time, I was unable to display this sentence in a unique color on the website.**

1. (15) Build a package.json file for Q2. If we run it, there should be no errors or warning when we try to install & run your code from #2 above. (You may assume yout application name is *Quiz1Server*)

**You can create a package file using init. You can install init through the command line (npm init). Please see the package.json file in my folder.**

1. (20) Explain *in detail* what the following code does; (also add comments to the code explaining what each line does)

var net = require('net') **// set variable net to require the net module**

var sockets=[]; **// set the variable sockets to an empty array**

var s = net.Server(function(socket) { **// event emitter that is used to create a local server**

sockets.push(socket); **// each socket is pushed into the array**

**// for when data is received**

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

for(var i=0; i<sockets.length;i++) { **// go through the sockets array**

if (sockets[i]==socket) continue; **// when going through the array, if the socket in array matches current socket, continue**

sockets[i].write(d); **// write to socket, return true if data flushed successfully to buffer**

}

});

**// for when other end of socket sends FIN packet**

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

var i=sockets.indexOf(socket);

**// destroy file descriptor**

sockets.splice(i,1); // or delete sockets[i]

});

});

s.listen(8000);

**This code creates a webserver in node.js that listens on port 8000. The server requires net, which is a modeule that provides the user with an asynchronous network wrapper. An empty array is created. Then, an event emitter is called that creates a local server. When data is received, you go through each element in the sockets array. If the element in the array matches your current socket, you write to the socket. When the other end of the socket sends a FIN packet, the socket will destroy its file descriptor.**