Web Science

Quiz 1: February 26, 2018

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-S18-Quiz1-*yourname*-quiz1.docx

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

ITWS4500-S18-Quiz1-*yourname*-*yourRCSID*

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

ITWS4500-S18-Quiz1-*yourname*-*yourRCSID*.zip

And submit it to LMS

1. **Frameworks** (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.

You can determine what type of device it is being displayed on through media queries. The size of the device can limit what is shown on the screen due to the size of the screen. This allows for better interaction with your page. Another reason you would consider doing this is enabling different types of interaction. For instance if it is shown on a touch sensitive device and you want to disable the ability to zoom due to whatever reason you can, or you can enable a different way to hover over things on touch devices.

* 1. (5) What is a package.json file? How is it used? What it used for? Is it required?

A package.json file is a file that tells npm what it needs to install to run this particular app. Whether it needs express or just node and what versions of each of these technologies you need to install to assure that it will work on any device capable of npm-ing the data. Most often it is auto-generated as you npm init and any npm installs after that are usually included. The require tabs in node servers can also enable a change in your package.json file. It is required to serve a node application locally.

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

NVM is node version manager. It is often used in conjunction with Git or Subversion to allow the user to keep up with (or reverse) node updates via nvm.

* 1. (10) Describe the difference between Front-end and Back-end frameworks. Provide at least 2 examples for each in your answer. (Be clear in your descriptions, ie ‘why is it back/front-end?’)

Front-end frameworks leverage and simplify building html and displaying data. Bootstrap and AngularJS are two different front-end technologies. They rely on dynamic changes, additions and subtractions to the html and can change what is displayed.

On the converse side, back-end technologies are server code and frameworks for building entire applications. These are never (or barely) seen by the client when they are connecting to the webpages. Two back-end frameworks are Express and RubyonRails. Applications run on these systems and they are rarely used for working directly with html.

1. **Node.js** : (40 points) Create a webserver in node.js, name your server *yourRCSid-Quiz1Server.js)* you may use express, but you *may not use a generator* – (ie NOT express-generator), which will serve a simple HTML page with an input field for zipcode and a button labeled ‘Run’ when a GET request is received on http://localhost:3000. Upon entering a zipcode and clicking the button, the server should get the current temperature for that zipcode and output a sentence that says the name of the location and whether it is Freezing (<=0C), Cold (btw 0 and 10), Warm (btw 11 and 25) or Hot (>25) – display the corresponding message in a unique color for each category. Allow the user to enter additional zipcodes for weather in other locations. Display the new sentence above the previous. Include a button that allows the user to refresh the page and start over.

1. (15) Build a package.json file for Q2. If we run it, there should be no errors or warnings when we try to install & run your code from #2 above. (You should make your application name : *yourRCSid-Quiz1Server.js*)
2. (20) Explain *in detail* what the following code does; (also add *stylized* comments to the code explaining what each line does, and highlight and correct any errors)

var net = require('net'); *// This is telling npm it requires this to run properly*

var sockets==[]; *// Creates an array called sockets*

*// the following code uses the Server property of net*

*// In this case s is a code that runs through socket connections*

*// and writes d to the socket when it has data*

*// If the socket ends the connection then s removes that socket // from its list*

var s = net.createServer(function(sock) { *// Creates server*

sockets.push(sock~~et~~); *// Adds sock to sockets*

sockets.on('data', function(d) { *// When we get data*

for(var i=0; i<sockets.length;i++) { *// find the sock*

if (sockets[i]==socket) continue; *// If it is, skip*

sockets[i].write(d); *// write data to all other // sockets*

}

});

sockets.on('end', function() { *// On end, run this // function*

var i=sockets.indexOf(sock~~et~~); *// find the index of this // sock*

sockets.splice(i,1); *// erase this sock*

});

});

*// the server is listening on port 8080*

s.listen(8080);