Web Science

Quiz 1: March 2, 2017

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-S17-Quiz1-*yourname*-quiz1.docx

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

ITWS4500-S17-Quiz1-*yourname*-*yourRCSID*

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

ITWS4500-S17-Quiz1-*yourname*-*yourRCSID*.zip

And submit it to LMS

1. **Frameworks** (25 points): (Answer in complete sentences, explain your answers)
   1. (5) What is MongoDB? How does it differ from MySQL (aka MariaDB)?
   2. (5) What is npm? How is it used? What it used for?
   3. (5) What is nvm? How does it work? Why is it used?
   4. (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 decriptions, ie ‘why is it back/front-end?’)
2. **Node.js** : (40 points) Create a webserver in node.js, using express – (NOT express-generator), which will serve a simple HTML page with an input filed and a button labeled ‘Run’ when GET request is received on <http://localhost:3000>. Upon entering a zipcode and clicking the button, the page server should get the current temperature for that zipcode and output a sentence that says 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. Include a button that allows the user to refresh the page and enter a new zipcode.

1. (15) Build an npm 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 your application name is *Quiz1Server*)
2. (20) Explain *in detail* what the following code does; (also add *stylized* comments to the code explaining what each line does)

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(8088);