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)?

**MongoDB is a database used for storing information in a document format. This is dissimilar to MySQL’s relational model in that it doesn’t require a predefined list of “columns” defining how your data should be entered. Instead, it is more similar to JSON objects where you define a number of key and value pairs that are stored. A document comprises these key and value pairs and a collection is similar to MySQL’s table in that it contains a set of documents. This flexibility in fields and structures is what sets MongoDB apart from MySQL.**

* 1. (5) What is npm? How is it used? What it used for?

**npm stands for node package manager. Just like how a developer can include libraries in a C++ or Java application, the same can be done in Node. These packages, or libraries, contain a plethora of methods and need to easily be included in a project. That’s where npm comes in. It is a method for making sure you have the libraries and their dependencies and an extremely streamlined way of installing the packages you want to use. It comes bundled with any release of Node and can be used to install packages with a command that’s simply *npm install.* In order to use this command, however, you need a file named *package.json*. This file is used to define the packages your project depends on and wants access to. When *npm install* is run, it goes through each dependency in the *package.json* and installs it AND each of that dependency’s dependencies (a mouthful I know) to your local project.**

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

**nvm stands for Node Version Manager and it allows for multiple versions of node to be running on a single machine. This is useful for testing applications across the many supported versions of Node that could be running on a server somewhere. It’s also useful when you have different projects on a single machine that disagree on which node version to use. It works by storing the binaries for the different versions of Node and updating the PATH environment variable to point at whichever version you want to use. For example, your system will have the PATH set to the default Node directory in /bin, maybe this version is 6.0.0. If you want to use version 0.10.0 for a project, all you have to do is install it using nvm and run *nvm use 0.10.0* and the node binary of v0.10.0 gets added to the PATH.**

* 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 decriptions, ie ‘why is it back/front-end?’)

**Frontend frameworks deal primarily with the views and interactivity of a web page. For example, this would be the placement and resizing of HTML elements using the framework Bootstrap or the dynamic binding of text to a variable using the framework Angular. All in all it deals with what the user would see on a webpage. On the flip side, backend frameworks help control how the page is served, how data is stored, or how the API works. For example, Express is a Node framework that controls the routing and serving of a web app and Ruby on Rails simplifies a lot of what goes into making a website as an API.**

1. **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')*//This imports the module*

var sockets=[]; *//handles an array of TCP connections*

var s = net.Server(function(socket) { *//this establishes a*

*connection to the server*

sockets.push(socket); *//Stores the newly created socket/conn.*

*// Every time data is received, send it out to every socket*

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

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

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

sockets[i].write(d);

}

});

*//when the socket connection is closed, remove the socket from the array of current connections*

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

var i=sockets.indexOf(socket);

sockets.splice(i,1);

});

});

s.listen(8088); *The network port the server is established*

**This code is utilizing a package called net, used for creating clients and servers and the ability to stream information from end to end. In this example it has created a data distribution service. Basically a bunch of connections, or sockets, connect to a single server. Every time data is sent from one socket to the server, that data is replicated to every socket that is connected to the server. Upon disconnecting, a socket is removed from this array of sockets receiving data from one another.**