Web Science

Quiz 2: May 9, 2016

100 points max

* Add your code in separate files in a folder/directory called rcsid-quiz2/. (You will zip them up in the end)
* Include your written answers in this file. Use a different font/color, in-line, in this document. Save it as rcsid-quiz2.docx and place in into the root of your quiz2 folder when completed
* When finished, zip your entire server (ie ../rcsid-quiz2/\*.\*) into a file named using the class convention (rcsid-ITWS4500-quiz2.zip) and upload it to LMS. Make sure you include any basic instructions into a readme.
* The quiz is due at 5:50pm or it will be late. After 6 it will NOT be accepted.

1. (60 points) Using node.js, create a server that when run will display a web page. Once the request comes to the server, the server should perform the following tasks;
   1. Serve a page to the user, welcoming them and which will inform them of what is happening
   2. On your page should be a button labeled, ‘go’
   3. When the user hits ‘go’, read from the zipcode file provided (you may edit the file as you like or put it into another format or database) [10 points]
   4. Use the open weather API to get the weather for each of the zipcodes in the file *as well as* the weather for your current location – which you are to retrieve using the geolocation API. [20 points]
   5. Read from the API into a mongo database. Your database should store (at least) [20 points]
      1. Zipcode
      2. Location name
      3. Current weather
   6. Once complete, let the user know that the database has been populated and show another button labeled ‘display’
   7. When display is pressed, display your current weather along with the zipcode, location and current weather of each zipcode in the file. [10 points]

Currently, I was unable to finish this. Right now the application asks for the users location. It then reads in the zip code file. It gets the current temp of each zip code. I set up the mongo database but didn’t have a chance to insert the information in it. I created a display button, but it is hidden because the function doesn’t complete yet.

1. (20 points) Build an foaf (Friend of a friend) file for yourself. This file should describe you (at least) by name, nickname and email. It should also list at least one person you know. Finally, it should indicate that you are a member of the group ‘RPI-ITWS4500’. Name it rcsid.foaf and place it in the root of your rcsid-quiz2 folder.
2. (20 points) Visualization: Answer in your own words and in complete sentences with proper grammar.
   1. Part 1: Pick two data visualization JavaScript libraries and compare and contrast them (focus on the differences) [10 points]
      1. Google Chart Tools allows you to display live data on your site. It was first released in 2007 and is rendered in HTML5 charts using SVG and VML. There are 13 chart types available in 2D and maps are also available as a GeoChart. Column charts, bar charts, line charts, area charts, pie and donut charts, scatter charts, stacked charts, pareto charts, stock charts and gauge charts are some of the charts available.
      2. Chart.js is an easy object oriented client side graphs for designers and developers. It was first released in 2013 and is rendered in canvas based charts. There are 6 chart types available. Line charts, area charts, pie and donut charts, spline charts, polar charts and radar charts are the only visualizations offered with Chart.js.
      3. Chart.js has limited features and only supports animation and a legend compared to Google Chart Tools which supports tool-tips, legend, multiple axis’s, number formatting, color palette and theming and trend lines.
      4. Both Chart.js and Google Chart Tools support mouse over interactivity and on click interactivity.
   2. Part 2: Which one of the two libraries you picked above would you use if you had to visualize tweets on a certain topic and why? (elaborate) [5 points]
      1. Due to its simplicity and clean look, I would use Chart.js for my basic charts. The library makes it very simple to create different graphs just off of an array of data.
      2. If I wanted to create a more complex chart, like a GeoChart (which is a map with data points plotted on it), I would use Google Chart Tools. Google Chart Tools allows you to have more variations in your visualization, but sometimes it is not needed.
      3. Depending on what I am doing to visualize tweets would determine which library I would use to visualize the information. Personally, I used both of these libraries while visualizing my tweets in Lab 10 so that I could provide clean simple charts while also showing tweets based on their location in the United States.
   3. Part 3: How would you pass the information (tweets and corresponding analysis) to your library of choice? Be specific. [5 points]
      1. For Chart.js graphs on the server side I create an empty array with each value initialized to zero. Then I create a query with the Twitter Streaming API and keep track of when each value is said in a tweet with my array. I pass the array back to the client side. On the client side I check to make sure the function is successful and parse my array. When I have all of the values I want to display I create a new chart with the information. Based off the default chart, I select what type of chart I want, the labels for my dataset, my array and the color(s) for the chart. I add in any options I want and then my chart is complete.
      2. For Google Chart Tools on the server side I create an object with arrays of information. Then I create a query with the Twitter Streaming API and keep track of longitude and latitude of each tweet as well as the user and text. I pass this object back to the client side. On the client side I check to make sure the function is successful and create a Google Data Table. I then add the columns of information I would need and set my options. After that I create the chart. I then parse through my data and for every value I want I put it in an array of arrays. I then add the overall array as a row in my Google Data Table, which adds it to the GeoChart.