Programming Assignment 1
Web Mashup: Display House Address on a Map with Weather details

Due on Tuesday June 21 before midnight
Worth 8% of your final grade

Description

The goal of this project is to create a web mashup that combines two web services: Google Maps and the Geoname, using JavaScript and AJAX. When you click on a on the map, your application will display the postal address and the weather.

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism.** That is, your program will be compared with the programs of the other students in class as well as with the programs submitted in previous years.

Note that, if you use a Search Engine to find similar programs on the web, we will find these programs too. So don't do it because you will get caught and you will get an F in the course (this is cheating). Don't look for code to use for your project on the web or from other students (current or past). Just do your project alone using the help given in this project description and from your instructor and GTA only. Finally, you should not post your code nor deploy your project on a public web site.

Platform

You will do this project on your own PC/laptop. You need to install the XAMPP web server, which includes the Apache http web server, PHP, MySQL (MariaDB), and PHPMyAdmin (these are the only components you need). It's about 125MBs (775MBs after installation) and can be installed on Windows, Linux, and OS X. The installation directory is \xampp for Windows, /opt/lampp for Linux, and /Applications/XAMPP for OS X. To start the server on Windows, you run \xampp\xampp-control.exe and you start the Apache web server. You may have to change the Security properties of this executable to Full Control for Users. If you get an error "Apache shutdown unexpectedly", read this.

You will test the project on your PC/laptop using either the Chrome or Firefox browser. The project grading will be done on a Firefox browser. If you do not have Firefox installed, you can install it from http://www.mozilla.org/en-US/. Firefox comes with Web Developer Tools for debugging HTML and JavaScript.

Setting up your project

Download the project1 zipped directory from blackboard. Unarchive the files inside your web server document root directory. The project1 directory contains 2 files: webmash.html and webmash.js. See the example in webmash.js. Your project is to edit webmash.html and webmash.js as explained in the description of the web application.

Web Services used by the Web Mashup

For this project, you will use the

- Weather Service from the GeoNames API (more specifically, findNearByWeatherXML)
- Geocoding and Reverse Geocoding from the Google Maps JavaScript API
- Google Map Markers

To use Google Maps, you need to get an API key (you will need to login using your google account). To use the GeoNames API, you need to register and obtain a username which will act as the API Key. After you get the two API keys, you put them in webmash.html and webmash.js (replace ???? with your keys), you start your XAMPP web server, and you test your setup on your web browser by using the URL address http://localhost/project1/webmash.html.

Project Description

You need to edit the HTML file webmash.html and the JavaScript file webmash.js. Your HTML web page must have 3 sections:

1. a button: Clear.
2. a Google map of size 600*500 pixels, initially centered at (32.75, -97.13) with zoom level 17
3. a text display area

Your program must insert an overlay marker on the Google map pinned on the latest address that displays the postal address and the weather (temperature, windspeed and clouds) from GeoNames. The text display area is the history log that displays all the addresses and its respective weather details that you have found so far (latest address is last). Each time you find a house, you erase the old marker from the map (if any), you display a new marker on the map on the house location (with address and weather), and you append this information to the display area.

To implement, when you click on the map, your program must find the address of the point you clicked (using Reverse Geocoding) and while clicking on map, you will get the latitude and longitude of the point which is sent to GeoName API (findNearByWeatherXML) for getting the weather details.

Note that the call to the findNearByWeatherXML API must be done using Ajax: inside the callback function (the listener for the left click) of the map, you should create an Ajax request that calls the findNearByWeatherXML API. When the result arrives (this is the callback of the Ajax request), you extract the temperature, clouds and windspeed and you display a new overlay marker on the map at the point you clicked. The overlay marker must display the postal address and weather details. The same information must be appended at the end of the text display area (third section). Note also that the map must display at most one marker and the text display area may contain multiple addresses/zestimates. If it is an invalid address or there is no weather details, you don't change anything. Finally, the Clear button clears the text input only.

Hints:

- How to URL-encode the address to mp it to GeoNames api: use the method

encodeURI(address).

Note that everything should be done asynchronously and your web page should never be redrawn completely. You need only one XMLHttpRequest object for sending a request to GeoNames, since Google Maps is already asynchronous.