

Assignment 3. Mobile Tracker of VGI

Due November 1st, 2023, at 11:59 EST

Worth 25 points

Using your assigned teams, you will create a web application that uses the GPS on your mobile device to track your team's movements while they have it open.

The crowdsourced data will be displayed on an online map. In this assignment we move away from Mapbox studio and use Mapbox GL JS within HTML.

We acknowledge that you may find solutions to writing apps online that utilize languages other than HTML/JS/CSS and that some of you may know other languages that can be used to develop on the web. Since this is a class for beginners, we consciously teach web development using HTML/JS/CSS only; consequently, those are the only languages which you are permitted to use when programming your app.¹ Code submitted in other languages will not be marked.

For this assignment you have been given a .php file to use. Since we are giving it to you and the assignment does not require you to modify it in any way, that rule above does not apply to that specific file. Don't worry about including it in your final submission.

Goals


- Learn how to combine hardware and web-based software tools to monitor and record users' VGI in space-time.
- Write your first Javascript (JS) "app" (technically webpages).
- Learn how to store data created in HTML/JS online - in this case as .csv and .json/.geojson
- Distinguish different types of crowdsourced contributions in a database.
- Learn how to display data on an online map.
- Individually reflect on the assignment.

Resources

- https://www.w3schools.com/html/html5_geolocation.asp
- https://www.w3schools.com/js/js_timing.asp
- https://www.w3schools.com/js/js_dates.asp
- <https://docs.mapbox.com/mapbox-gl-js/example/simple-map/>

¹ You can use jquery if you like.

Tasks

1. Build a data collection app. You need to build an app that allows a user to collect location data from their phone on an interval (“live tracking” works this way). This is accomplished through JS and relies on several factors including the phone’s location service hardware (usually a GPS). You will do this with JS embedded in an HTML page. This data collection app needs to:
 - A. ask the user to enter (or select from a list/options) their user ID. Look up Javascript “prompts” to accomplish this. Write some logic that limits the user ids which can access the page (e.g., only users who are in your team and whose ids are stored in a list).
FYI: Your User Id must not contain spaces and only alphanumeric (A-Z, a-z and 0-9).
 - B. display some typical things like an app title, logo/picture, and descriptive text. The best apps show creativity, which is not the same as aesthetics. Most important is that your app functions. Somewhere on the page, display the user ID that was entered in the first step. (e.g., “You are logged on as: USER1”)
 - C. Take a screenshot of your user interface (i.e., what the user sees).

 - D. collect the path travelled by the user while the app is open and the tracker is enabled.² It should get locations of the user’s phone and the date/time for those locations every 30 seconds while the page is open.

When you open your app your browser will ask you to give it permission to share your location. You must click “yes” for the app to work. If for some reason you click no, you may have to go into your browser security settings and re-allow the url of your app to ask for location information. If this happens feel free to reach out to us for help.
2. Store the data. Your app needs some place in the “cloud” to store your data. If it was stored on your phone alone then you would not be able to see anyone else’s data. As soon as you close your phone’s browser, the data would disappear. To accomplish this, you will use code that creates a .csv file on the neogeoweb.ca server. Use the assignment4.zip package for this next part.
 - i. In your website code, you will need the function ‘writeToCSV’. We wrote this for you. It is in the assignment4.zip package.
 - ii. Add the “writeToCSV” function into the JS section of your app.

² Websites only collect locations while the page is open and active. This is a security measure that is built into the HTML framework that we use. If the page is closed or you lock your phone screen, the app will stop recording.

- iii. Save the formatAsJson.php file into the same folder as the app.
- a. Every 30 seconds, send your data to this function. Code this in JavaScript.

For Your Understanding: the programming language PHP, unlike Javascript, can easily read and write external files. Here we use it to write our datasets to .csv and .json files.

How it works:

The “writeToCSV” function will send your collected data to the formatAsJson.php file as often as you call the function. When it receives the data, formatAsJson.php saves your collected data to some files that it will create: data.csv and json files for each user (i.e., user1.json, user2.json).

data.csv is a csv record of all entries to the dataset
<USERNAME>.json are geojson files which can be read by the Mapbox API as line features.

These files will automatically appear the first time you run the function, and update accordingly.

BE CAREFUL: if your variables are wrong these files will not generate in the correct format, and you may have to delete them and run the code again.

- 3. Crowdsourcing location data. Instruct the user to load the app and track themselves for a while as they move around the city - this will require a smart phone and data plan. Make sure they show movement. We will not accept data that only shows a stationary user (lots of dots all in the same area). In the past creative groups have had friends in other cities and relatives in other countries open the app and track themselves. You can track yourself as you walk around campus but that is not the most interesting dataset. At minimum track yourself as you move around the City of Montreal.
 - a. There is no limit to the amount of data you collect, but you should plan on at minimum tracking a user for at least one hour total of movement.
 - b. A user can access the app to track themselves on multiple occasions to meet the tracking time goal but they should always be moving when they are tracking themselves.
 - c. It is a good idea to check the .csv on the server occasionally to see that it is receiving the data from the phone. If not, there is something wrong with your code.


In this assignment we do not use Mapbox studio to store and/or style our data like in Assignment 1 and 2. You MUST create a Mapbox GL JS standalone Webmap like the ones on the example pages (<https://docs.mapbox.com/mapbox-gl-js/example/simple-map/>).

You may, however, use Mapox Studio to design publish a custom style (background map) for use in your final app (e.g., style: 'mapbox://styles/itatters/ck5mtgi2q1ymx1linkl6r0n1nh', //hosted style id)

4. Map all the users' paths. Use Mapbox GL JS for your mapping app.
 - a. Create a second app, with the needed JS code to embed (mashup) the map.
 - b. In that app, you will need to input the data collected by your users and have it display on the map. Use the <USERNAME>.json files and Mapbox GL JS to achieve this. The paths need to be in different colours to represent each user.

Ian wrote a function for importing the geojsons into Mapbox that you could use:

```
//add a geojson(json) layer to the map
//replace everything within the < > appropriately
map.on('load', function() {
//the url (name) of the .json file - must be in same folder as the
javascript var url = <USERNAME>.json';
//add a source
    map.addSource('<SOURCE NAME>', { type: 'geojson', data: url});
    map.addLayer({
        id: "<THE USER'S NAME>",
        type: "line",
        source: "<SOURCE NAME>"
    })
});
});
```

- c. Take a screenshot of the map that contains the users' coloured data paths. 
 - d. Augment the user interface. Add a toggle so one can choose which user's data to display. Add popups for the lines, so when they are clicked on information is conveyed.
5. Use .css or .html templates to make your pages look cohesive (i.e., so that they look like part of a single app) and attractive. You could ask GPT to write this part for you. Provide a hyperlink so that you can switch to the other app.
6. Upload all the files, including data.csv and the <USERNAME>.json files, your two apps (tracker and map) and the writetocsv.php file in the same folder in your team's account on neogeoweb.ca.

7. Document your application in a report. That same report needs to include your individual reflections. In a word document your team should provide the following information:
 - a. Describe the workflow in your application (~1 page, which excludes screenshots but could include a diagram).
 - b. In approximately one page describe the process of collecting and sending the locations from your phone.
 - c. In no more than one page describe what purpose the php & csv hold in this application.
 - d. In approximately one page describe the process of setting up Mapbox and mapping the locations from your phone.
 - e. Include a user id so we can test the app.
 - f. Don't forget to mention whether you used a GPT and how.

Individually (i.e., each person) reflect on the assignment (~0.5 pg/person)

- f. What was the most challenging part of the assignment for you?
- g. Were you surprised at how easy it was to collect a user's space-time signature? Why or why not? Draw on material discussed in class to answer these questions.
- h. As part of your surprise (or not), does the tracker keep working if your phone's screen is off? Or if you close your browser? What about if you are not using the browser on your phone: is the app still running somewhere in the background?

Submission

You need to submit your two apps and a report.

1. Email us your report as a single .doc/team, including the screenshots (renee.sieber@mcgill.ca and sichen.wan@mail.mcgill.ca) and individual reflections.
2. Upload your apps to a new folder in your group's directory on the neogeoweb.ca server (you will need to do this anyway because it is important for your data collection and mapping apps).
3. Include the final URLs of both the data collection app and the mapping app in your email text.