G1 Progress Report 1

In the past month, I worked on setting up the working environment, installing the required software and tools, and study the technologies that are needed. I use Visual Studio Code as my main text editor and set up the command-line tools: iterm, zsh, and bash. I installed PostgreSQL, PostGIS, Jupyternote book, and Esri Python Rest API by downloading from their official websites or using Anaconda. The technologies I have studied include Asynchronous JavaScript, Bootstrap, Git and GitHub, PostgreSQL, and API.

Setting up the environment is a crucial part of the project that needs a lot of patience. It builds the foundation for the effective and efficient work of the project. I copied the source code of the AFV project to my local Mac and worked on it using VS code and the Live Server in it. I pushed the new version to GitHub using Git for version control. Now I have three commits: original, mobile version, and collaborative version. The configuration on the MacBook is different than some of these on the Windows OS, so I switched from Bash to zsh because some of the command lines did not work in Bash when I tried to set up the PostgreSQL. However, the PostgreSQL works smoothly in zsh. I mostly use the command line for the database query, but sometimes I use a Mac Postgres graphic interface to connect to the database and visualize the server setups.

I made some justification for my project. Although I planned to use bootstrap to make the interface more mobile-friendly in June, I just applied it to the JavaScript code last week. I made the change because I need more support from the previous group to finish the other parts of the project. For example, I need more information to connect to the original database so I would be able to work on the data using the python script. While I was waiting to get the original scripts and database dumps from the group members, I worked on the Bootstrap modification, which can be done on my own.

The Bootstrap makes the website more responsive and mobile-friendly. The examples in Figure 1 show the original website on the left side and the small screen version on the right side. The display on the left screenshot is in proper layout, but when the screen is narrow, in which I simulate the display on a smartphone, the layout of the text looks squeeze, the yellow-background <div> looks very narrow, and there is a blank section on the bottom-right corner. In Figure 2, the modified version using bootstrap looks more proper in its display. When the screen is narrow, the sections of the <div> stack on top of each other, and the text looks properly arranged. I also use the Bootstrap library to change some of the colors and fonts of the text to make it a more modern look.

A close up of a map

Description automatically generatedA screenshot of a map

Description automatically generated

Figure1. The original website and the display on a small screen.

A close up of a map

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Figure 2. Modified website using Bootstrap and its display on a small screen.

Here are some details and suggestions I noticed when using Bootstrap. One can use either CDN or complied CSS and JS file of Boostrap. The CDN seems handier and satisfies most use cases. Assuming we are designing a website for mobile phone users, “col-sm-\*” class for grid division for small devices should be applied. I may need to keep separate versions of codes for mobile and laptop page because the sizing system for bootstrap is specifically made by using xs, sm, md, and lg class. I changed the mapcanvas class to 500px instead of 100%, or the map will disappear because of some display conflict.