Developing a Map with links to files

1. Gathering the links to the files

The files are stored in Google Drive thus, retrieving the links to all those files is the foremost task. And having many files make it impossible to do by hand (one file link at a time).

So, we write a script code for the same.

```
function myFunction() {

var ss=SpreadsheetApp.getActiveSpreadsheet();

var s=ss.getActiveSheet();

var c=s.getActiveCell();

var fldr=DriveApp.getFolderById("REPLACE_[ID]_HERE");

var files=fldr.getFiles();

var names=[],f,str;


while (files.hasNext()) {

f=files.next();

str='=hyperlink("' + f.getUrl() + '","' + f.getName() + '")';

names.push([str]);

}


while (files.hasNext()) {

f=files.next();

names.push([f.getUrl(), f.getName()]);

}

s.getRange(c.getRow(),c.getColumn(),names.length, 2).setValues(names); }
```

Go to Google Drive and get the shareable link of the folder where all your files are saved. Now, in the end of that link there is an "ID"

For e.g.

This is the link

https://drive.google.com/drive/folders/1BbJ3ARWDs3Ivu9AoPLmSrpgwxoxgRlRz?usp=sharing

And for this link following is the script code

```
function myFunction() {


var ss=SpreadsheetApp.getActiveSpreadsheet();

var s=ss.getActiveSheet();

var c=s.getActiveCell();

var fldr=DriveApp.getFolderById("1BbJ3ARWDs3Ivu9AoPLmSrpgwxoxgRlRz");

var files=fldr.getFiles();

var names=[],f,str;


/*

while (files.hasNext()) {

f=files.next();

str='=hyperlink("' + f.getUrl() + '","' + f.getName() + '")';

names.push([str]);

}

*/


while (files.hasNext()) {

f=files.next();

names.push([f.getUrl(), f.getName()]);

}


s.getRange(c.getRow(),c.getColumn(),names.length, 2).setValues(names); }
```

Now open a New Spread Sheet in your Google Drive and go to Tools>Script Editor
and paste the script code in it and press "Run"(play like button).

We will get the links and file names in the Google Sheet and download this data.

There are many locations for whom the files are not present thus we need to arrange the file links according to their names.

The latitude-longitude database with their names and all details are present in the file on Dropbox, download this data.

Now, we need to merge these 2 data files. Using **VLOOKUP** function in **Microsoft Excel** arrange the links with their names.
The resulting data sheet is all we wanted regarding the information to be displayed on the map.
So far, we have used some scripting code and basic excel functions to extract data.

Proceeding to the core stuff (making the webpage with the map)

2. Developing a webpage with a Map

The Google Map API is free to use for non-professional purposes.
First step would be:

a. Getting an API KEY
   https://developers.google.com/maps/documentation/javascript/get-api-key
   Refer to the above link for getting your own API KEY.

b. Placing a Marker on your Map
   https://developers.google.com/maps/documentation/javascript/adding-a-google-map
   Refer to the above link for understanding the functions used in making a map and placing a marker.

c. Adding an info window to be displayed for each marker
   https://developers.google.com/maps/documentation/javascript/infowindows
   Refer the above link for implementing an info window.

Now we know all the basic concepts we needed.

Moving forward we want multiple markers and different info window for each of them. Good thing here is that we already have all the raw data we needed.

We are writing a PHP code to develop the map needed, the code is **m1** file:

To put data in it there are 2 codes which convert the data (location + info window) into the PHP code.

Run these two codes and give the data as input; paste the output in the PHP code (marker location + info window content).

Now the PHP code is good to deploy. [GitHub link to all the needed codes]

Whenever you click any marker an info window will popup with a link to the data related to that location.

For any further support drop an e-mail:

adityakr.hurkat.min17@iitbhu.ac.in