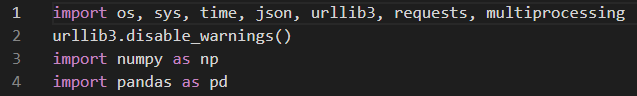


**DOWNLOADING WEATHER DATA FOR MULTIPLE POINTS USING**

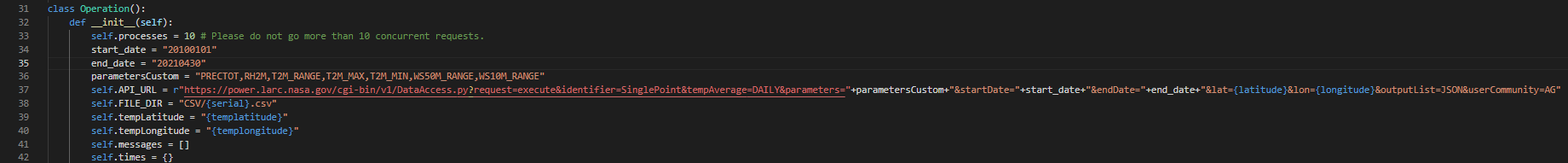
**PYTHON**

1. **Program Explanation**

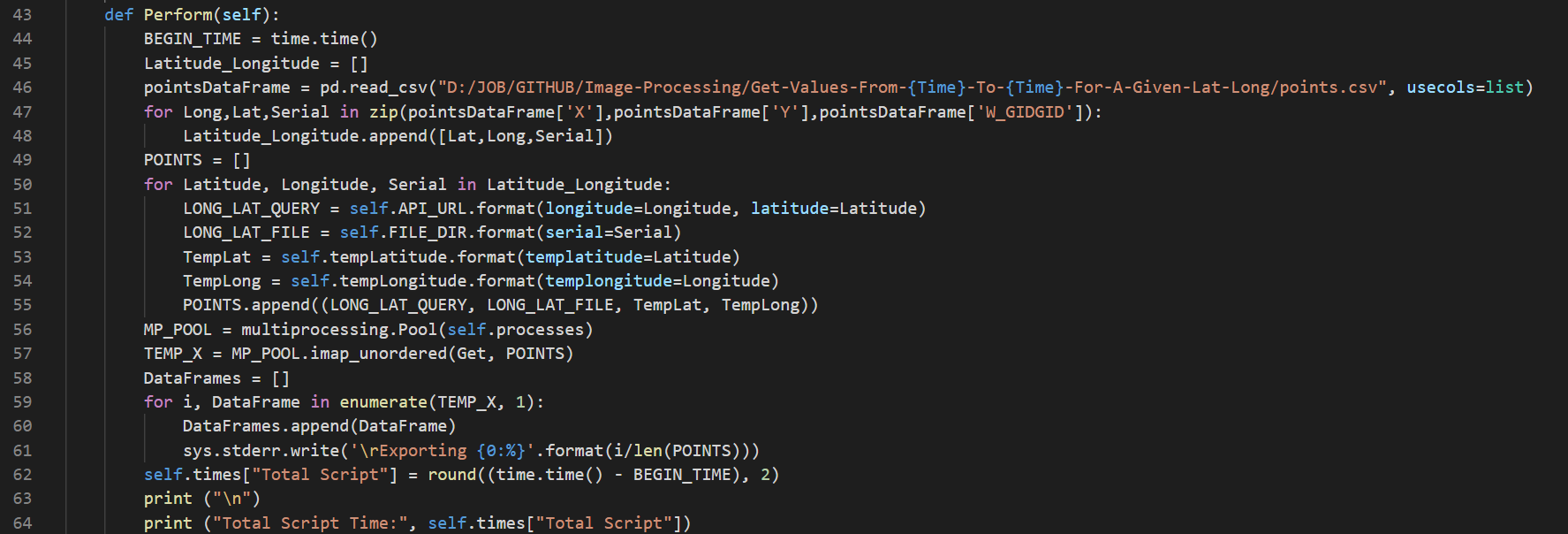
**1.1 Weather Data Code Explanation**



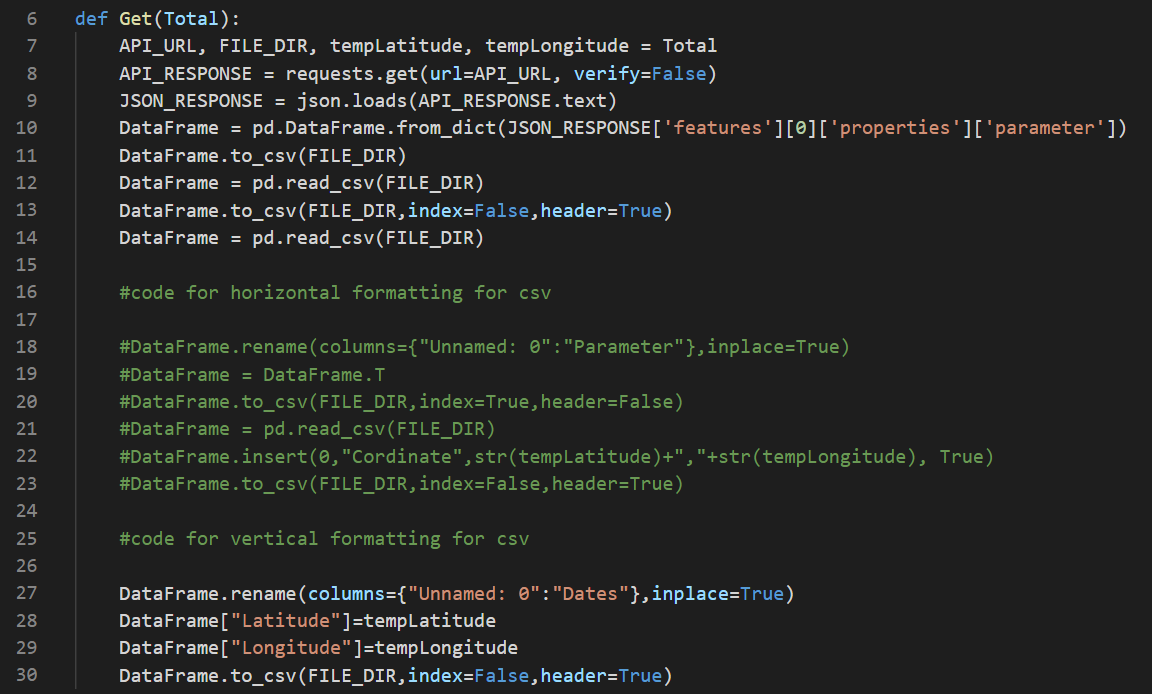
* We will be importing libraries os, sys, time, json, urlibr3, requests, multiprocessing, numpy and pandas



* Create a class Operation() with constructor.
* In constructor method we will be defining variables process, start\_date, end\_date, parametersCustom, API\_URL, FILE\_DIR, tempLatitude, tempLongitude, messages and times
* The process variable defines how many times the code should access the API\_URL maximum should be only 10, if exceeded it will give an error saying too many requests from API\_URL.
* The start\_date and end\_date parameters are for the user to define the date range of the data (s)he wants. Should be same format as specified.
* The parametersCustom variable defines how many parameters of data should the code download which should be defined by the user. Should be the same format defined in the website.
* API\_URL variable have the url to the api to get data with the parameters defined by user added within the url passed to that variable.
* FILE\_DIR variable holds the directory where the downloaded csv files should be.
* tempLatitude and temp Longitude have the latitude and longitude(points specified in the csv file)
* Messages and times are used for Perform() method for error message storing.

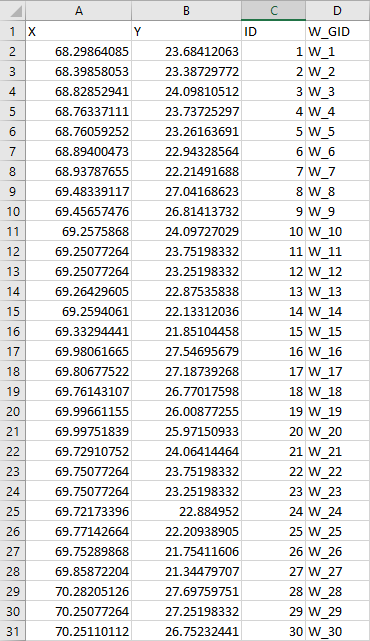


* Now define a method Perform() in the same class, the BEGIN\_TIME variable records the current time while the performing the operation and it will be used to determine the time taken for the download.
* Latitude\_Longitude variable is an array which stores the latitude and longitudes given in the csv input file.
* pointsDataFrame variable holds the data of a csv file which will be given by the user.
* For more clarification we will be looping that data frame and take out the only data we were gonna need, which will be the latitude, longitude and the W\_ID.
* Those 3 variables will be appended to the array we created earlier, every time it loops it will add one row from the dataframe until the end of rows.
* Now we will use the data from the Latitude\_longitude array and extract the 3 variables (latitude, longitude, W\_ID) and start the loop for each latitude, longitude and W\_ID respectively.
* We will be passing the latitude and longitude to the variable we created earilier in the constructor API\_URL.
* And we will be passing the W\_ID as the serial number of the csv files which are being created.
* Same for the variables Templat and Templong aslo formatted with latitude and longitude respectively.
* All this data will be appended to the array called points, each row will be appended while the loop continues.
* MP\_POOL variable will be set to the processes that were define in the constructor earlier.
* TEMP\_X will have the out data by passing the GET method and the points array in it.
* After the data we will be calculating the live percentage which should be printed on the screen will downloading the data using for loop through the dataframe of the csv file we read earilier.



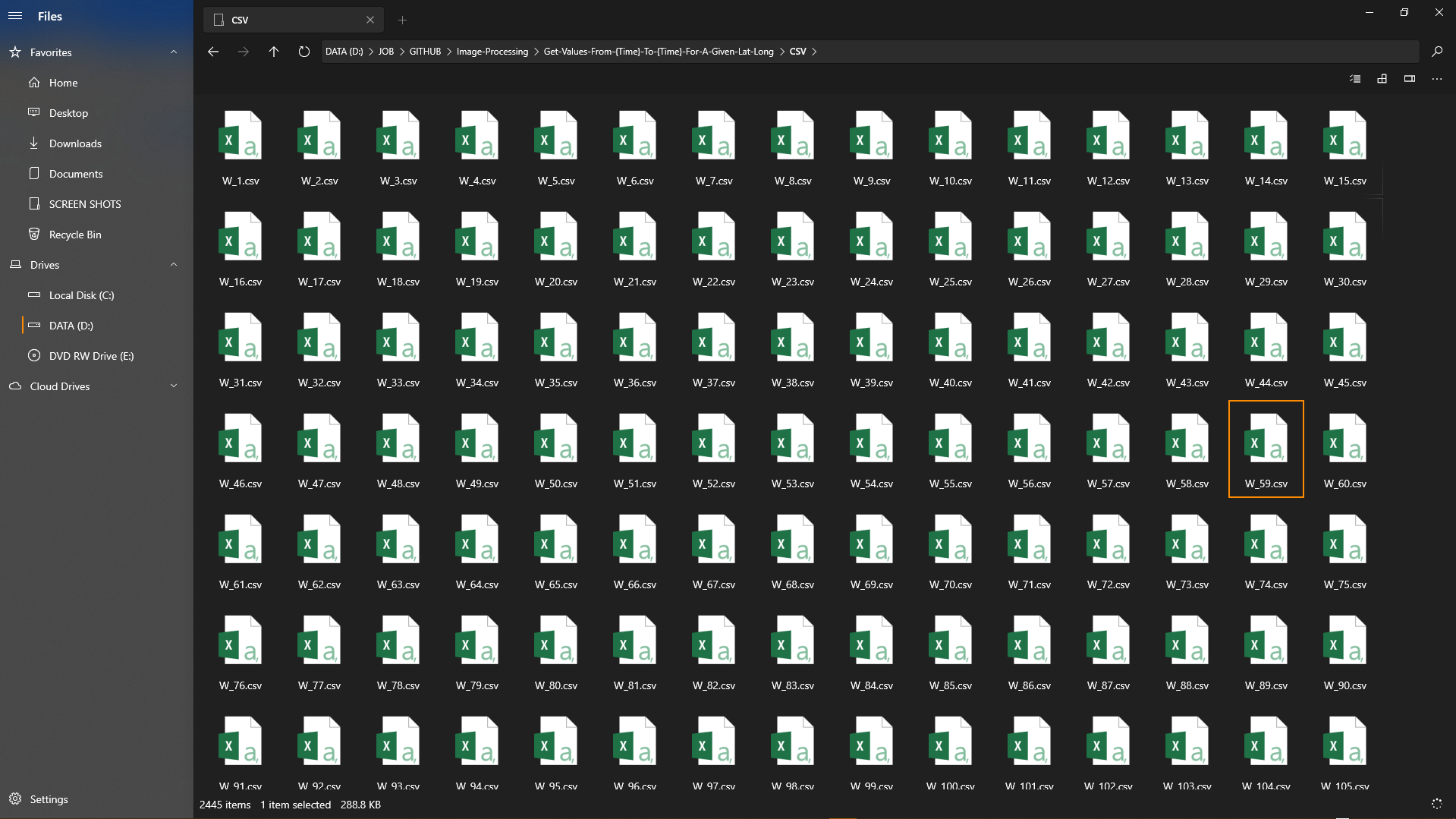
* Outside of the class Operation() we will be defining a method called Get(), which has been called in the method Perfrom().
* This method consists of variable API\_URL, FILE\_DIR, tempLatitude, tempLongitude which will be defined to a variable total which consists of 4 types of variable in it which makes it equal to be define it with another 4 variables.
* API\_RESPOSE gets the data the we need by passing the API\_URL variable into the requests using the requests library.
* JSON\_RESPOSE variable will possess the data that have been retrieved by the API\_RESPONSE variable and loads it in plain text which will be easy to be read or manipulated.
* DataFrame variable stores the data from the dictionary of JSON data which have been retrieved and stored as plain text
* After that the DataFrame will be will save to a csv file and read again into the same variable
* Then the DataFrame variable will be added a header and removed the index and saved to csv again
* Again the csv will be read into the same variable now the columns will be renamed using a function called rename from pandas library.
* And 2 columns will be added Latitude and Longitude with latitude and longitude points then again saved to the cv file
* This process will repeat for every row read from the given csv file.

**1.2 Input Data Format**



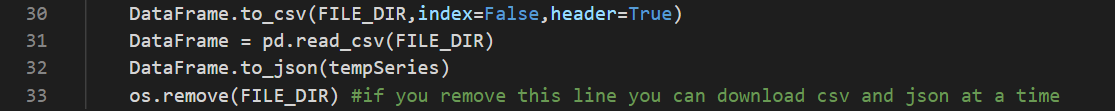
* The csv file which is going to be included in the code should at least contain 3 columns in which 2 columns are Longitude and Latitude respectively and the other column should be the W\_ID.
* Failed to provide the columns as is should result error as output.

**1.3 Output Data**



* For every point one csv file will be created with the parameters and date range specified by the user.

**1.3 For JSON as output**



* A small 3 lines of code will be added which reads the output csv and converts into JSON and removes the csv this way without writing a whole code for JSON output these 3 lines does the trick.