**Code Documentation**

1. WeeklyReport.gs

* **function onOpen(e){} :**

function onOpen(e) {

SpreadsheetApp.getUi()

.createMenu('Run')

.addItem('Get Data', 'getDateWise')

.addToUi();

}

* onOpen is a trigger function which is used to create a custom menu for our Spreadsheet.
* When clicked on the Get Data menu in the sheet, **getDataWise()** function is called.
* **function getDateWise(){}:**
* This is our main function where we fetch the required data from the sheet and compare the values of user entered Dates.
* This main function uses two sub-functions for data calculations (filtering, sorting or arithmetic tasks) :
  1. function getUploaderRegionData(fdRange, tdRange){}
  2. function filterAndCount(arr, bc){}
* **function getUploaderRegionData(fdRange, tdRange){} :**

function getUploaderRegionData(fdrange,tdrange)

{

var chss = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("CH ID-Summay");

var dtrgn = chss.getRange(fdrange+":"+tdrange).getLastRow();

var rawDataRegion = chss.getRange(fdrange+":"+tdrange).getDisplayValues();

var six = 0;

var i=0;

var uniqueUploaders =[];

while(i<dtrgn)

{

for(var j=0;j<rawDataRegion.length;j++)

{

uniqueUploaders.push(rawDataRegion[j][i]);

}

i+=6;

}

var hellooo = filterAndCount(uniqueUploaders);

var clist = hellooo.list;

var ctotal = hellooo.total;

var ccount = hellooo.count;

return clist;

}

* This function filters the entire data range to get the value only from the dates mentioned by the user (fdRange, the first parameter and tdRange, the second parameter).
* The filtered data region is then pushed into an array and returned as **clist.**
* **function fAndC(arr,tsbc) :**

function filterAndCount(arr) {

var total ,list = [], count = [], prev;

arr.sort();

for ( var i = 0; i < arr.length; i++ ) {

if ( arr[i] == prev ) {

count[count.length-1]++;

}

else {

list.push(arr[i]);

count.push(1);

}

prev = arr[i];

}

var total = arr.length;

return {

list: list,

count: count,

total:total

};

}

* As the name specifies, this function iterates over the data (arr, the first parameter) to check for a specific uploader (tsbc, the second parameter)
* And hence returns the list of filtered data.

1. TimeSpent.gs

* **function onEdit(e){}:**

function onEdit(e)

{

var tsss = SpreadsheetApp.getActiveSpreadsheet();

var tss = tsss.getActiveSheet();

var excludes = ["Template","Login ID","CH ID-Summary","Weekly Summary"];

if (excludes.indexOf(tss.getName()) !== -1) return;

var tscell = e.range;

tsdoSomething(tscell);

}

* + - This is a trigger function for getting the changed status value of a specific cell.
    - Here, the ‘excludes’ is a list of sheets name where we do not want to perform the trigger function.
    - In between this trigger function we are calling **tsdoSomething(tscell)** where tscell is the changed cell value.
* **function tsdoSomething(tscell){} :**
* This is the main function of the sheet where we get all our work done.
* First of all it gets all the value of some specific cells by using simple pieces of code such as:

var tsloginTime = tssheet.getRange("C2").getDisplayValue();

* This main function uses three sub-functions for data calculations (filtering, sorting or arithmetic tasks) :

function fAndC(arr,tsbc){}

function tsgetHourDiff(a, b){}

function isValidHour(hour) {}

* **function fAndC(arr,tsbc) :**

function fAndC(arr,tsbc) {

var tslist = [],tsprev;

for ( var i = 0; i < arr.length; i++ ) {

if ( arr[i][0] === tsbc && arr[i][3]!=="" ) {

tslist.push(arr[i]);

}

}

return tslist;

}

* fAndC stands for **filter and count**.
* As the name specifies, this function iterates over the data (arr, the first parameter) to check for a specific uploader (tsbc, the second parameter)
* And hence returns the list filtered data.
* **function tsgetHourDiff(a, b){}**

function tsgetHourDiff(a, b) {

if (!isValidHour(a) || !isValidHour(b)) {

return "Invalid input(s)";

}

var h1 = a.split(":"), h2 = b.split(":");

var h = 0, m = 0;

h = h1[0] - h2[0];

m = h1[1] - h2[1];

if (h < 0) {

h = -h;

m = -m;

}

if (h == 0) {

m = Math.abs(m);

}

if (m < 0) {

m = m + 60;

h = h - 1;

}

return m;

}

* This function gets two hour values (a and b) and thus calculated the time difference between them.
* And hence returns the time difference ‘m’ (in minutes).
* **function isValidHour(hour) :**

function isValidHour(hour) {

hourPattern = "^([0-1]?[0-9]|2[0-3]):[0-5][0-9]$";

if (hour.match(hourPattern)) {return true;}

return false;

}

* This function simply checks if the data entered by the user is of a specific format by comparing it with a regular expression.