```php
<?php

require '../vendor/autoload.php';
include ("../include/login.php");

use PhpOffice\PhpSpreadsheet\IOFactory;
use PhpOffice\PhpSpreadsheet\Spreadsheet;


$id = $_GET['id'] ? $_GET['id'] : 0;
$utcOffset = $_GET['utc'] ? $_GET['utc'] : "+0";

$query = "SELECT * FROM logs
INNER JOIN logtypes ON logtypes.logtype_id = logs.logType
WHERE log_id = '".$id."'";

$result = $mysqli->query($query) OR DIE($mysqli->error);
$logRow = $result->fetch_assoc();

$startTime = new DateTime($logRow['date'], new DateTimeZone("UTC"));

// Create new Spreadsheet object
$spreadsheet = new Spreadsheet();

// Add some data
$spreadsheet->setActiveSheetIndex(0)
    ->setCellValue('A1', 'Date')
    ->setCellValue('B1', $logRow['date'])
    ->setCellValue('A2', 'Notes')
    ->setCellValue('B2', $logRow['notes'])
    ->setCellValue('A3', 'Weather')
    ->setCellValue('B3', $logRow['weather']);


$query = "SELECT * FROM entries WHERE log_id = ".$id." ORDER BY time";
$result = $mysqli->query($query) OR DIE($mysqli->error);

while ($row = $result->fetch_assoc()) {
    $data[] = array('time' => $row['time'], 'count' => $row['count']);
}

// Parse the data into minute intervals
$intervals = (getTotalByInterval($data,60));


$spreadsheet->setActiveSheetIndex(0)
    ->setCellValue('A5', 'Minute Starting')
    ->setCellValue('B5', 'Count')
    ->setCellValue('C5', 'Running Total');


$startingRow = 6;
foreach($intervals as $k => $interval) {

    $currentRow = $startingRow + $k;

    $spreadsheet->setActiveSheetIndex(0)
    ->setCellValue('A'.$currentRow, $interval['time']->format('h:i'))
    ->setCellValue('B'.$currentRow, $interval['count'])
    ->setCellValue('C'.$currentRow, $interval['runningTotal']);
}
```

```
62
63
64
65  // Set active sheet index to the first sheet, so Excel opens this as the first sheet
66  $spreadsheet->setActiveSheetIndex(0);
67
68  // Redirect output to a client's web browser (Xlsx)
69  header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
70  header('Content-Disposition: attachment;filename="01simple.xlsx"');
71  header('Cache-Control: max-age=0');
72
73  $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
74  $writer->save('php://output');
75  exit;
76
77
78
79
80
81
82
83  function getTotalByInterval($data, $interval) {
84
85      $runningTotal = 0;
86
87      // Get the interval counts
88      $intervals = getCountByInterval($data, $interval);
89
90      // Iterate through the intervals
91      foreach($intervals as $k => $interval) {
92          // Add the interval's count to the running total
93          $runningTotal += $interval['count'];
94          // Add the running total to each interval
95          $intervals[$k]['runningTotal'] = $runningTotal;
96      }
97
98     return($intervals);
99  }
100
101
102 function getCountByInterval($data, $interval) {
103
104     $intervalData = array();
105
106     // Replace the time differential with datetime objects
107     $data = array_map('processDbData', $data);
108
109     // Get the starting time by rounding the first time down to the nearest interval
110     $firstTimestamp = $data[0]['time']->getTimeStamp();
111     $roundedTimestamp = floor(($firstTimestamp / $interval)) * $interval;
112     $currentInterval = new DateTime();
113     $currentInterval->setTimeZone(new DateTimeZone("UTC"));
114     $currentInterval->setTimestamp($roundedTimestamp);
115
116     $currentCount = 0;
117     $i = 0;
118
119     while (count($data) > 0 && $i < 10000) {
120
121         $currentIntervalEnd = clone $currentInterval;
122         $currentIntervalEnd->modify("+".$interval." sec");
123
```

```
124         // If the entry is in the current interval add it to the count
125         if ($data[0]['time'] < $currentIntervalEnd) {
126             $count = array_shift($data)['count'];
127             $currentCount += $count;
128         }
129 
130         // If the entry isn't in the current interval
131         else {
132             // Push the previous time / count to the intervalData array
133             $intervalData[] = ['time'=>clone $currentInterval, 'count'=>$currentCount];
134 
135             // Increment the current interval by the interval time and reset the count to 0
136             $currentInterval->modify("+".$interval." sec");
137             $currentCount = 0;
138         }
139 
140         $i++;
141 
142     }
143 
144     $intervalData[] = ['time'=>$currentInterval, 'count'=>$currentCount];
145 
146     return $intervalData;
147 
148 }
149 
150 function processDbData($row) {
151 
152     global $startTime;
153 
154     // Create a new date object
155     $dt = new DateTime();
156     $dt->setTimeZone(new DateTimeZone("UTC"));
157 
158     // Set the date object to the starttime unix timestamp + the time differential
159     $dt->setTimestamp($startTime->getTimeStamp() + ($row['time'] / 1000));
160 
161     // return an array with the new date
162     return ['time'=>$dt, 'count'=>$row['count']];
163 
164 }
165 
166 
167 
168 
169 
170 ?>
```