**PHP/MySQL API**

Actions.php

**-Increment(Boolean)**

Increment the foot traffic count of an individual entering or exiting. Takes one parameter of Boolean type

that represents entering (true), or exiting (false) respectively. Default true on empty parameter.

**-ResetDb()**

Deletes all foot traffic data from database and resets auto increment.

-**GetAllCounts()**

Returns the number of items labeled as entering or leaving within the table.

**-AnalzeQuery(String)**

Processes given SQL query passed as a string parameter, to return the given SUM, AVG, MIN, MAX for a given time interval for entering and leaving counts.

**-GetHourData(String)**

Passes given hour to AnaylzeQuery to return the given SUM, AVG, MIN, MAX for the specific time interval.

**-GetDayData(String)**

Passes given day to AnaylzeQuery to return the given SUM, AVG, MIN, MAX for the specific time interval.

**-GetWeekData(String)**

Passes given week to AnaylzeQuery to return the given SUM, AVG, MIN, MAX for the specific time interval.

**-GetMonthData(String)**

Passes given month to AnaylzeQuery to return the given SUM, AVG, MIN, MAX for the specific time interval.

**-CalculateMode(integer array)**

Calculates and returns the mode of the given integer array.

**-CalculateMedian(integer array)**

Calculates and returns the median of the given integer array.

Partials.php

**-HtmlHeader(String)**

Dynamically sets html header of every webpage based on given page title, imports necessary javascript/CSS/Google chart utilities.

**-HtmlNavbar(String)**

Based on string parameter displays the relative CSS to highlight the current page on the navigation bar, also handles all functionality of the navigation bar.

**-StandardChartForm()**

Prints standard chart form to allow specification of given time interval.

**-AnalysisTable()**

Prints standard table of Mean, Median, Mode, Max, Min, and Count for the specified given time interval.

monthData.php

* **GetMonthData($startDate, $endDate)**

Prints the count for number of pedestrians registered as exiting and entering for the user-specified time-interval (month).

* **List($result, $resultCounts);**

Prints the results of invoking GetMonthData in a list format.

* **json\_encode(String)**

Convert a data object to a string that is encoded in JSON format.

hourDara.php

* **GetHourData($selectedDate)**

Takes in a one date, particularly today’s date, and prints out the hour data which shows the count for number of pedestrians registered as exiting and entering.

* **List($result, $resultCounts);**

Prints the results of invoking GetHourData in a list format.

* **json\_encode(String)**

Convert a data object to a string that is encoded in JSON format.

dayData.php

* **GetDayData(startDate, $endDate)**

Takes in a time interval and prints out the day data which shows the count for number of pedestrians registered as exiting and entering.

* **List($result, $resultCounts);**

Prints the results of invoking GetDayData in a list format.

* **json\_encode(String)**

Convert a data object to a string that is encoded in JSON format.

weekData.php

* **GetWeekData(startDate, $endDate)**

Takes in a time interval and prints out the week data which shows the count for number of pedestrians registered as exiting and entering.

* **List($result, $resultCounts);**

Prints the results of invoking GetWeekData in a list format.

* **json\_encode(String)**

Convert a data object to a string that is encoded in JSON format.

LoadDayData.js

**-LoadDayGraph(integer array, function formatDate)**

Creates graph based on the array of data received from database broken into time intervals, formatDate

edits the axis of the graph to represent the specified interval relative to the array.

* **drawChart()**
  + **$.ajax()**
    - refers to the dayData.php specified by the relative path in the url section.
    - calls function(results) on success to draw the chart by loading google charts API package

LoadHourData.js

**-LoadHourGraph(integer array, function formatDate)**

Creates graph based on the array of data received from database broken into time intervals, formatDate

edits the axis of the graph to represent the specified interval relative to the array.

* **drawChart()**
  + **$.ajax()**
    - refers to the hourData.php specified by the relative path in the url section.
    - calls function(results) on success to draw the chart by loading google charts API package

LoadWeekData.js

**-LoadWeekGraph(integer array, function formatDate)**

Creates graph based on the array of data received from database broken into time intervals, formatDate

edits the axis of the graph to represent the specified interval relative to the array.

* **drawChart()**
  + **$.ajax()**
    - refers to the weekData.php specified by the relative path in the url section.
    - calls function(results) on success to draw the chart by loading google charts API package

LoadMonthData.js

**-LoadMonthGraph(integer array, function formatDate)**

Creates graph based on the array of data received from database broken into time intervals, formatDate

edits the axis of the graph to represent the specified interval relative to the array.

* **drawChart()**
  + **$.ajax()**
    - refers to the monthData.php specified by the relative path in the url section.
    - calls function(results) on success to draw the chart by loading google charts API package

3rd-Party Code disclosure

<https://momentjs.com/> - Used for formatting date

<https://developers.google.com/chart/> - Used to create visual charts

<https://getbootstrap.com/docs/4.0/getting-started/introduction/> - Used to create basic CSS headers/drop down navigation