Introduction

This report aims to provide the technical information required to maintain and implement the *Incident Clearance Time Prediction Program* and the *Incident Queue Length Prediction Program*.

Start Page

* Click to Start.html
  + This file’s primary purpose is to redirect the user to incident-clearance/incident.html
* incident.html
  + Allows users to choose between the *Incident Clearance Time Prediction Program* or the *Incident Queue Length Prediction Program*

*Incident Clearance Time Prediction Program*

* incident-clearance/highway\_type.html
  + Allows the user to choose which highway the incident occured on
  + JS inside the HTML file highlights the selected highway, and unselects all other selected highways
  + “Other Highways” leads to clusters.html
  + Clicking “Next” leads to tab.html
* tab.html
  + The name of the highway is taken from the postfix parameter after tab.html
    - For example, tab.html?md200 indicates to the program that md200 is the highway being studied
  + tab.html contains the HTML containers for the tabs that the user clicks through when they interface with the program
  + The tab switching is handled by tab.js, which is lined to tab.html on line 18 of tab.html
* tab.js
  + tab.js provides the button handlers which handle the tab switching
  + tab.js also provides the prediction algorithm for calculating the estimated clearance time, which can be found implemented in function updateTime()
  + updateTime() takes the incident information from var model, decides which situation applies, and calls drawSVG()
  + There are four drawSVG() methods, used to draw the three red bars in the prediction area of the program, and write the estimated time

*Incident Queue Length Prediction Program*

* incident-clearance/queue.html
  + This HTML file provides the containers that the user will use to navigate through the requirements for calculating an estimated queue length on a MD highway
  + The navigation through the HTML tabs is handled by queue.js
* queue.js
  + This JavaScript file handles the logic for the buttons, queue length calculation, and drawing the mean queue length bars
  + The “Save” button only becomes active when the proper data is inputted into the program, and this button enabling/disabling is handled with $("input").change
  + The calculateResults() function handles the queue length calculation logic
  + The printResults() function