

Intelligent Road Traffic Control (IRTC)

Prototype.

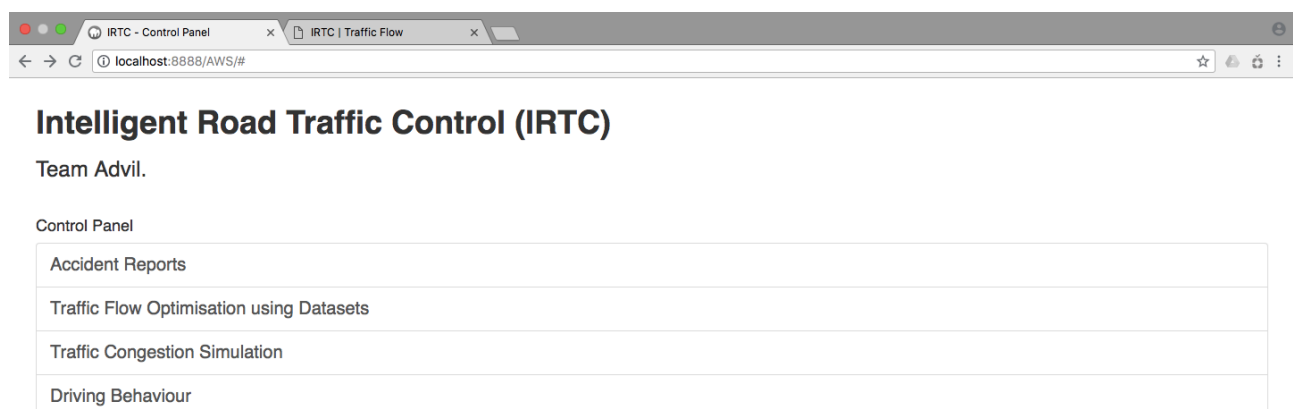
Cities across India are growing rapidly, thousands of vehicles are being added to the road everyday and the need of the hour is a technology for managing the road traffic. Providing citizens with accurate information on traffic conditions can encourage journeys at times of low congestion, and uptake of public transport.

For solving the problem, we have developed an application that will analyse and estimate the demands of road traffic users and calculate the timings of the traffic lights grid.

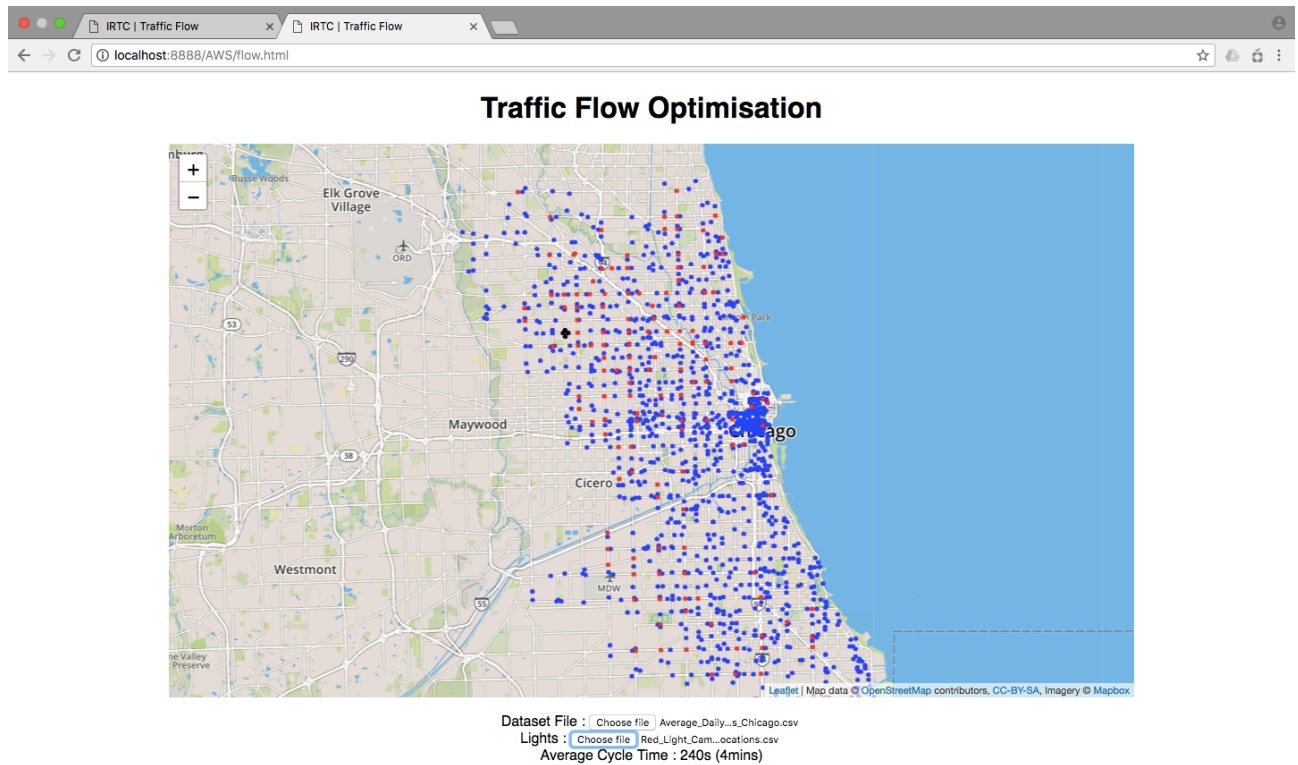
The prototype has the following implementations -

- 1) Traffic Flow optimisation using public datasets.
- 2) Congestion detection (simulation)
- 3) Accident Reports
- 4) Driving Behaviour analysis.

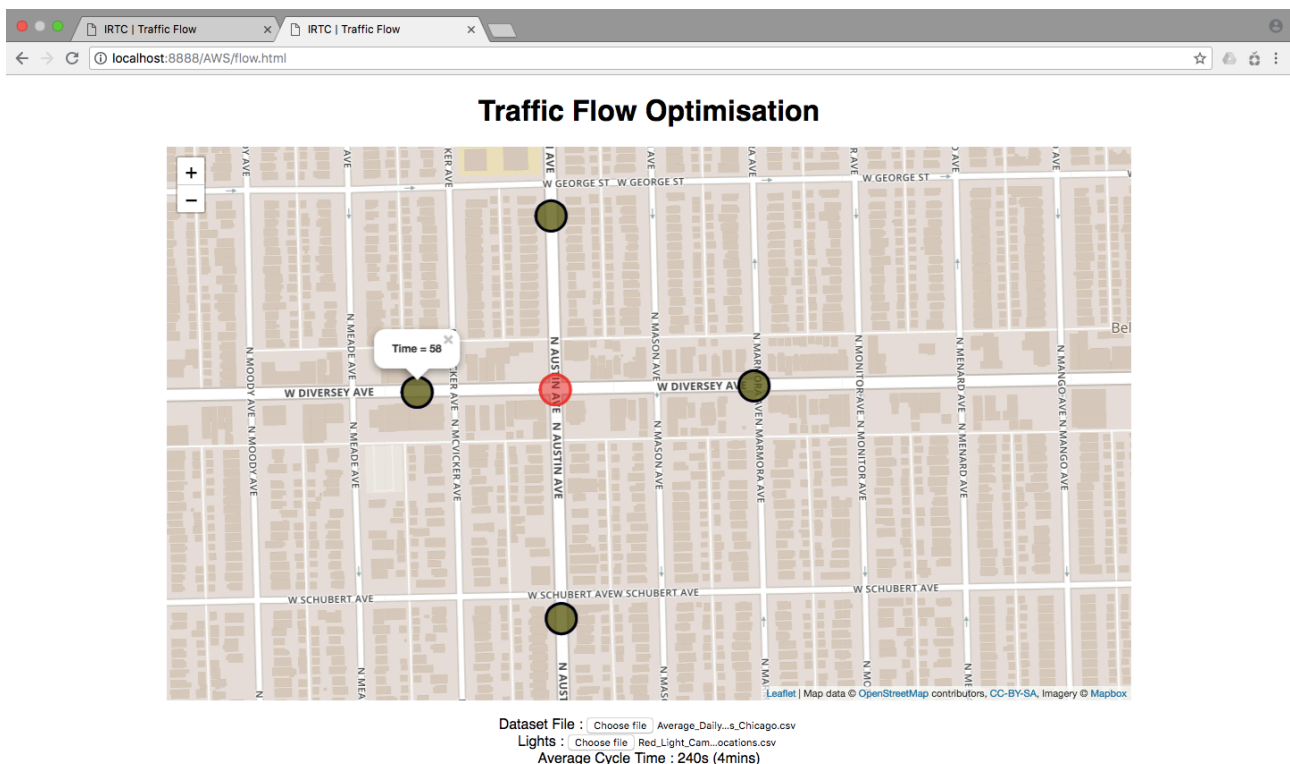
Dashboard (web)-



1) Traffic flow optimisation using datasets -



We have used two public datasets provided by IL, USA Government. The data is provided by the sensors that measure the total volume of traffic passing over them. These sensors are projected as blue color dots on the map. The data is of the city of Chicago. We have the locations of traffic signals and the sensors. Traffic signals are projected as red dots on the map.



From all the data provided, we have used a particular junction for demonstration.
If the average cycle time is kept fixe for instance 4 mins = 240s.

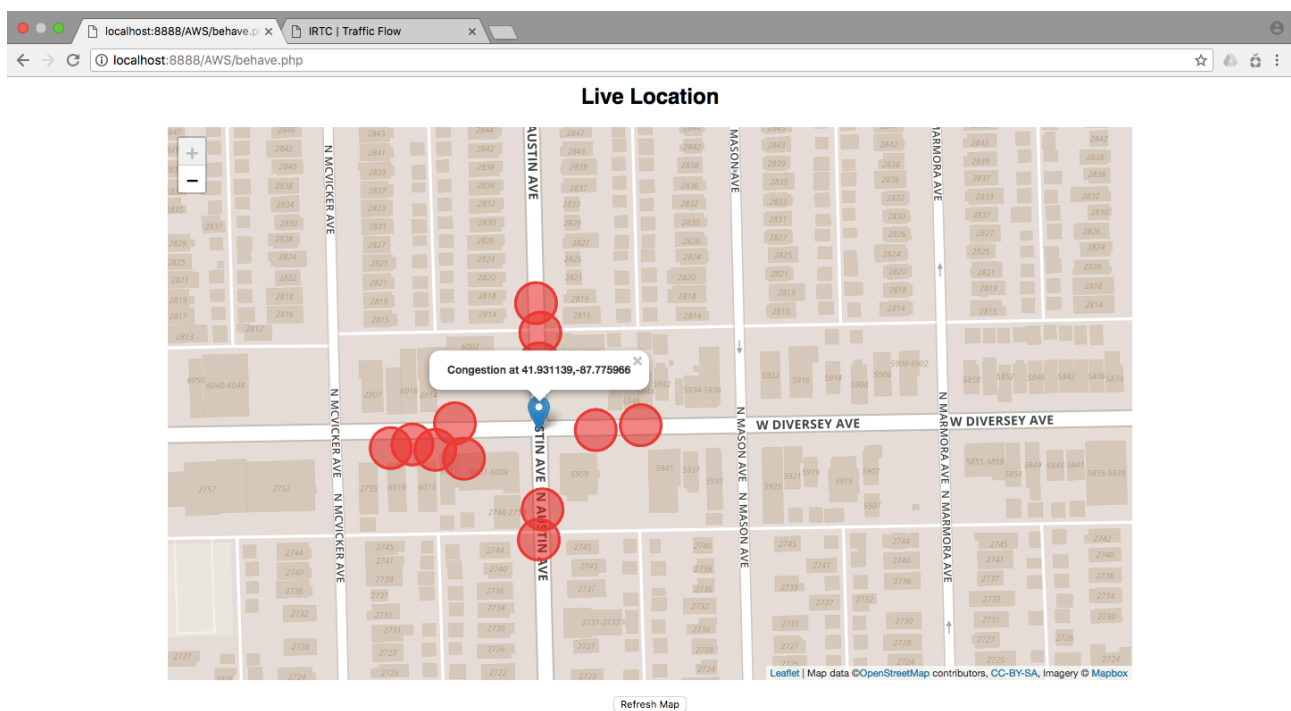
We can calculate the green signal time for each direction proportionally with the formula -

Green signal time from Direction ' X ' to the Junction ' J ' = Average Cycle Time *
(Volume passing through that direction / total volume passing through that junction) *
Driving behaviour factor.

We have used the driving behaviour factor to quantify the users driving behaviour at the junction. Irregular speeds, irregular accelerations are a characteristics of bad driving behaviour

Example calculations are given in the below pictures -

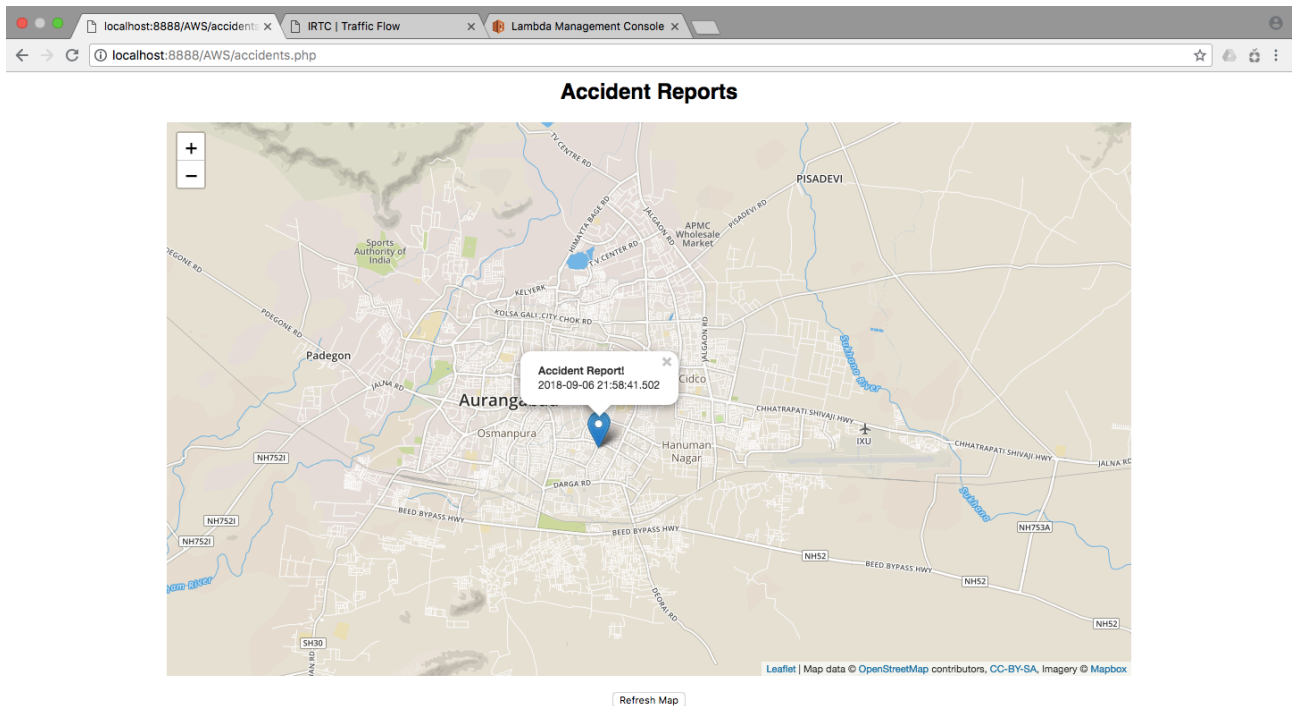
2) Congestion detection



The congestions can be detected in a map by detecting high density unique user points, we have created a table in the DynamoDB to store unique location.

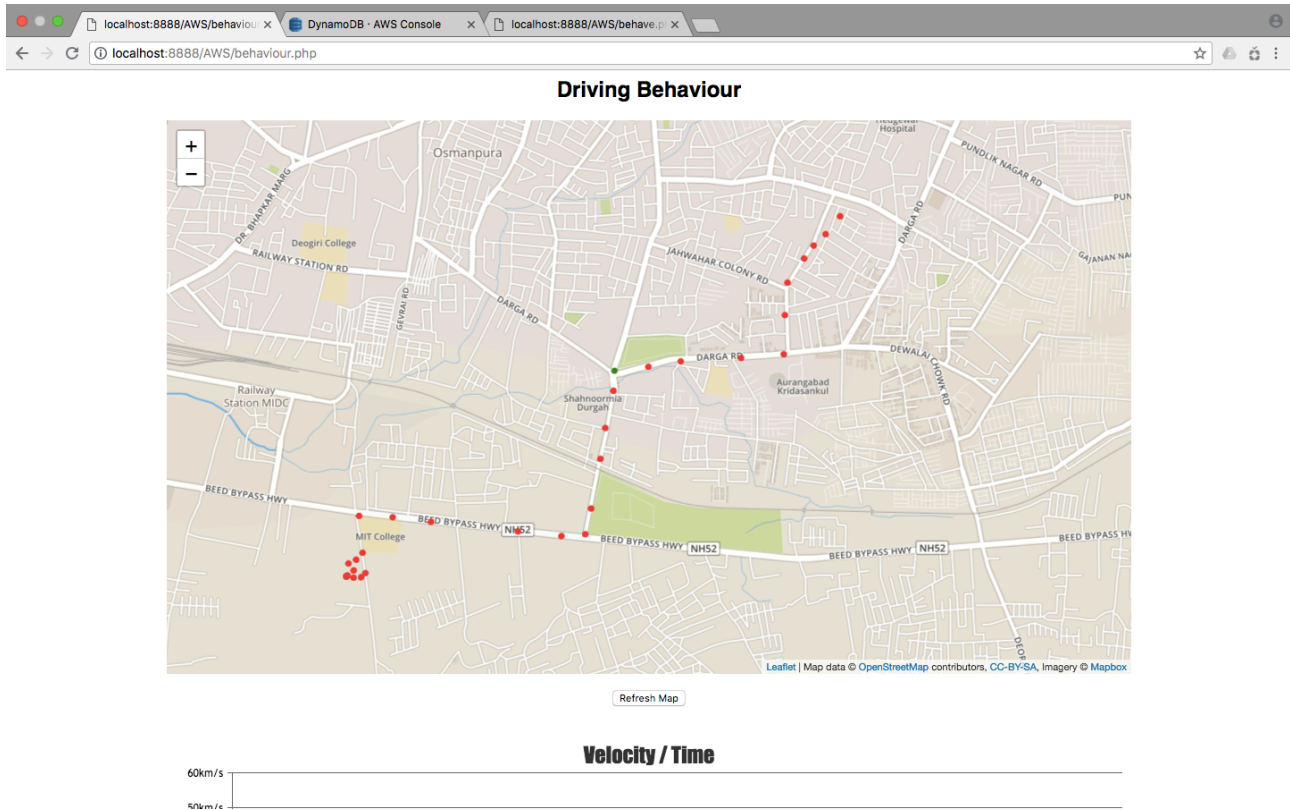
The red points denote unique users, higher their density, more the congestion.

3) Accidents can be reported from the mobile "IRTC" App with a press of button and will be shown on the dashboard.

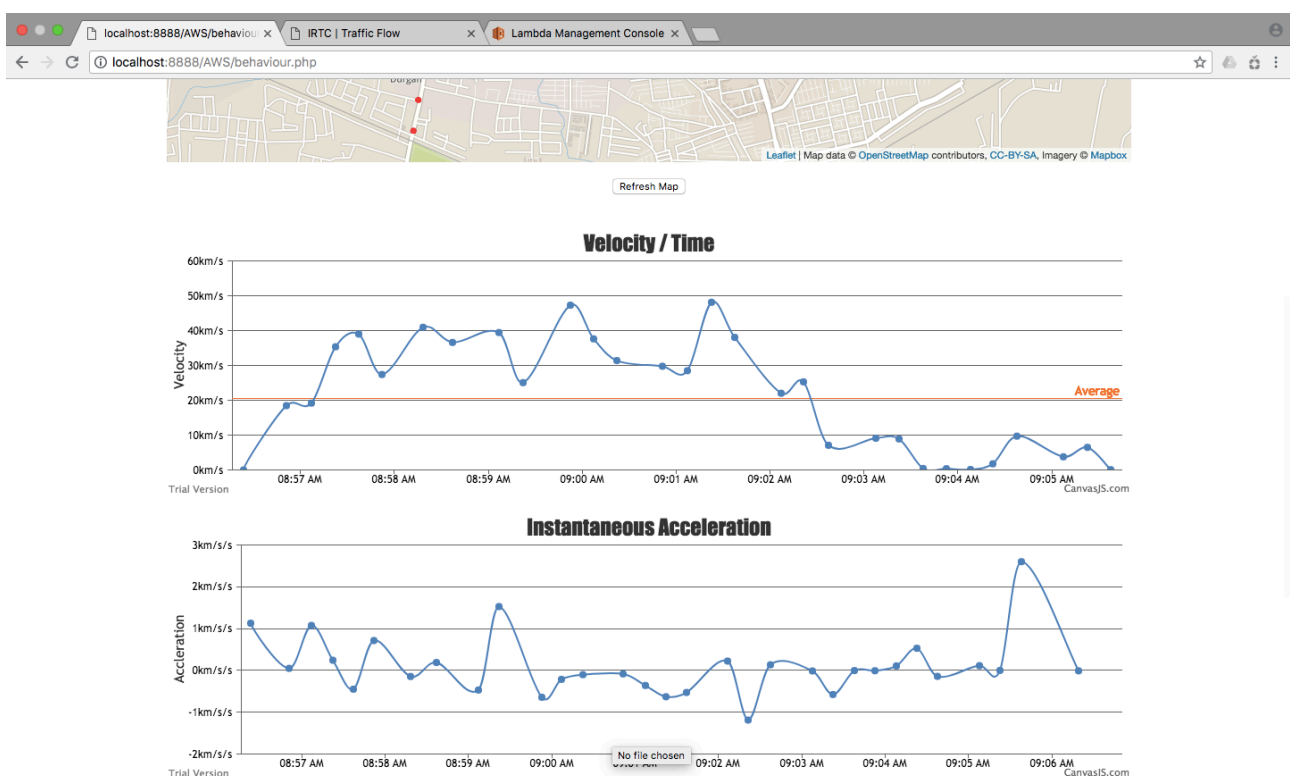


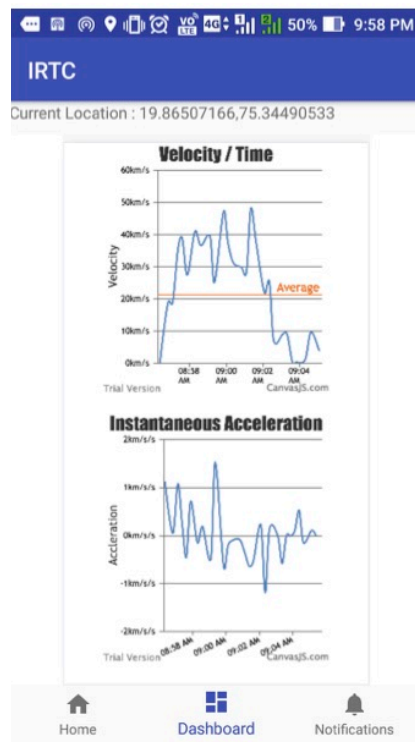
4) Driving Behaviour -

The User provides continuous location for the IRTC app in the background, anonymously. This data of the user will be analysed to understand the driving behaviour of the user. It involves Velocity patterns and Acceleration patterns in this prototype in the form of Graphs



The red points in the above image shows location points while driving.





Screenshot of the app showing driving behaviour.

THE APP

The mobile app will have three different classes of users -

- 1) Emergency Vehicles (Ambulance, Fire Fighters, Police, etc)
- 2) Public Transport (Buses, Trams, etc.)
- 3) Regular User.

The users, their features and functionalities -

- 1) Emergency Vehicle : The app will take input for a path from the Emergency vehicle user, when in emergency, the Traffic lights in that path will be optimised for smooth movement of the emergency vehicle. The nearby users in the traffic will be notified about an emergency vehicle arriving near them and ask them to give way for the same.
- 2) Public Transport : The app will analyse this class of vehicle with its daily route, using Amazon machine learning and optimise the traffic light timings for them. For instance, If a Bus is running late on its regular route, the bus will have less waiting time on the traffic signals.
- 3) Regular User : The regular user will provide the traffic data. Which will help our system to analyse the traffic density and congestions in realtime. The user can also

send images of congestions and defects in road (eg. potholes) at several locations. The media generated will be Stored in Amazon S3. The user can also notify about the

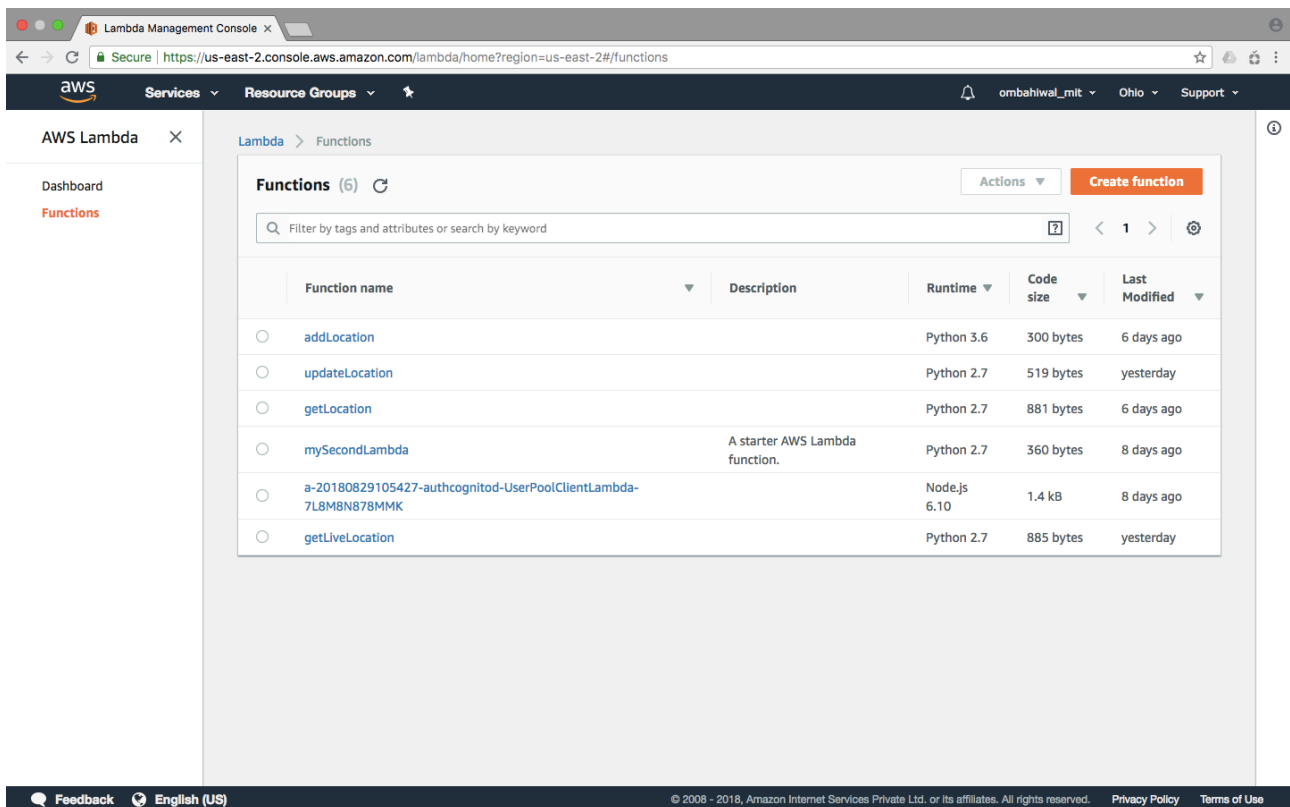
accident spots on their nearby locations. The authorities will be informed with this data in order to take proper action. The waiting time of the user will decrease on the traffic signals. According to the traffic density.

Overall, this system will help us in

- 1) Reducing traffic jams.
- 2) Optimising peak hour demands.
- 3) Optimise Public Transport Timings.
- 4) Facilitate emergency vehicles.
- 5) Help Identify Accident spots.
- 6) Understand user driving behaviour.
- 7) Reduce Economic loss.
- 8) Evaluate congestion spots and help in taking decisions to build new infrastructure.

USE OF AWS -

LAMBDA



The screenshot displays the AWS Lambda Management Console interface. The left sidebar shows the 'AWS Lambda' menu with 'Dashboard' and 'Functions' options. The main content area is titled 'Functions (6)' and includes a search bar and a 'Create function' button. Below this is a table listing six functions with columns for Function name, Description, Runtime, Code size, and Last Modified.

Function name	Description	Runtime	Code size	Last Modified
addLocation		Python 3.6	300 bytes	6 days ago
updateLocation		Python 2.7	519 bytes	yesterday
getLocation		Python 2.7	881 bytes	6 days ago
mySecondLambda	A starter AWS Lambda function.	Python 2.7	360 bytes	8 days ago
a-20180829105427-authcognitod-UserPoolClientLambda-7L8M8N878MMK		Node.js 6.10	1.4 kB	8 days ago
getLiveLocation		Python 2.7	885 bytes	yesterday

The footer of the console shows 'Feedback', 'English (US)', and copyright information: '© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

localhost:8888/AWS/accident... IRTC | Traffic Flow Lambda Management Console

Secure | https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions

Services Resource Groups

ombahawal_mit Ohio Support

AWS Lambda

Dashboard Functions

Lambda > Functions

Functions (6)

Filter by tags and attributes or search by keyword

Function name	Description	Runtime	Code size	Last Modified
addLocation		Python 3.6	300 bytes	6 days ago
updateLocation		Python 2.7	519 bytes	yesterday
getLocation		Python 2.7	881 bytes	6 days ago
mySecondLambda	A starter AWS Lambda function.	Python 2.7	360 bytes	8 days ago
a-20180829105427-authcognitod-UserPoolClientLambda-7L8M8N878MMK		Node.js 6.10	1.4 kB	8 days ago
getLiveLocation		Python 2.7	885 bytes	yesterday

localhost:8888/AWS/behav... IRTC | Traffic Flow DynamoDB · AWS Console

Secure | https://us-east-2.console.aws.amazon.com/dynamodb/home?region=us-east-2#tables:selected=unique_locations;tab=items

Services Resource Groups

ombahawal_mit Ohio Support

DynamoDB

Dashboard Tables Backups Reserved capacity Preferences Preview

DAX Dashboard Clusters Subnet groups Parameter groups Events

Create table Delete table

Filter by table name

Name

lrtc_db

unique_locations

unique_locations Close

Overview Items Metrics Alarms Capacity Indexes Global Tables Backups Triggers More

Create item Actions

Scan: [Table] unique_locations: user_id

Viewing 1 to 14 items

Scan [Table] unique_locations: user_id

Add filter

Start search

user_id	current_location
1	19.86507482,75.34489453
2	19.852268,75.2972352
3	41.931081, -87.776722
4	41.931065, -87.776853
5	41.931057, -87.776585
6	41.931017, -87.776413
7	41.931712, -87.77598
8	41.93158, -87.775954
9	41.931444, -87.775959
10	41.931145, -87.775621
11	41.931167, -87.775351

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

DYNAMODB

localhost:8888/AWS/behav... IRTC | Traffic Flow DynamoDB · AWS Console

Secure | https://us-east-2.console.aws.amazon.com/dynamodb/home?region=us-east-2#tables:selected=lrtc_db;tab=items

Services Resource Groups

ombahawal_mit Ohio Support

DynamoDB

Dashboard Tables Backups Reserved capacity Preferences Preview

DAX Dashboard Clusters Subnet groups Parameter groups Events

Create table Delete table

Filter by table name

Name

lrtc_db

unique_locations

lrtc_db Close

Overview Items Metrics Alarms Capacity Indexes Global Tables Backups Triggers More

Create item Actions

Scan: [Table] lrtc_db: lrtc_id

Viewing 1 to 37 items

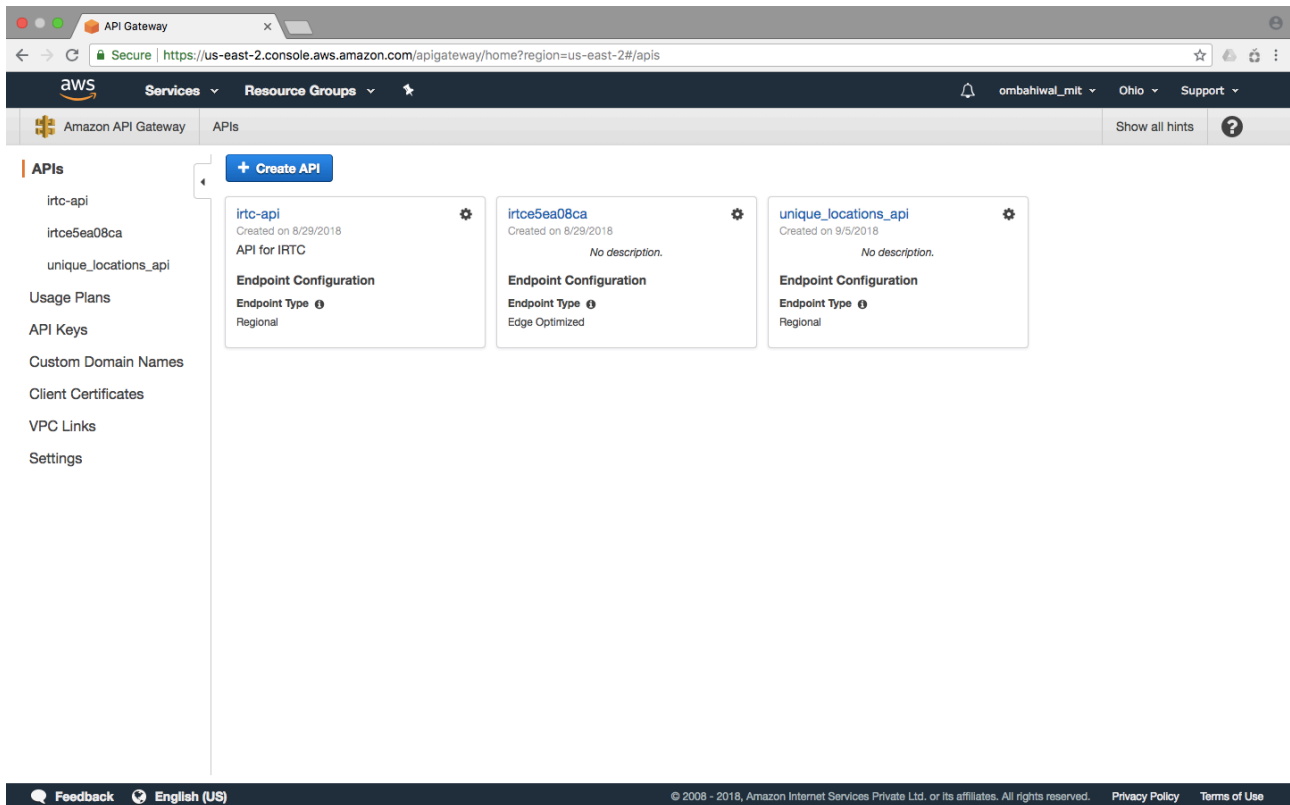
Scan [Table] lrtc_db: lrtc_id

Add filter

Start search

lrtc_id	location	status	timestamp
1705	19.85848798,75.33763585	0	2018-09-05 08:59:07.245
4858	19.84984906,75.32235232	0	2018-09-05 09:03:07.966
5850	19.85145151,75.32379571	0	2018-09-05 09:02:21.604

API GATEWAY



<https://github.com/ombahiwal/Irtc-aws-prototype>