Abstract

India boasts of being the second-largest road network in the world. The total stretch of the Indian road networks stand at a whopping 5.4 million km! Hence, it forms a huge ultimatum for the Indian Government to provide flawless roads at every step. For any Indian, be it average or millennial, driving through the Indian streets is certainly a hassle that no would like to go through.

Traffic blockage happens when a volume of traffic or modular split produces interest for space more prominent than the accessible road limit; this point is usually named immersion. There are various explicit conditions which cause or disturb clog; the greater part of them diminish the limit of a street at a given point or over a specific length, or increment the quantity of vehicles required for a given volume of individuals or products. Some of these mainstream traffic problems include-

- Poor road quality due to excessive traffic- The extreme congestion of urban roads due to heavily used private vehicles leads to the degradation of the quality of the roads. This leads to incessant traffic problems most of the times.
- Noise and Air pollution especially in urban areas- The sheer magnitude of traffic problems also gives rise to other health-damaging issues such as the air and sound pollution.
- Fixed signal timers- The traffic signal timers have a fixed time period to switch traffic between different directions which creates an issue even if the traffic density is less.

The current traffic control systems (TCS) in the metro cities of India are inefficient due to randomness in the traffic density pattern throughout the day. Previous works discussed in the literature review section, include systems based on image processing and sensors. However, the issues with these works is that they disburse unwarranted capital and are limited in their capabilities for inputs. Along with that, these systems also require complex arrangements and take up needless space which could otherwise be very well avoided.

So, we propose a solution which controls the traffic dynamically based on various important factors like time of day, climate, condition of roads etc. This system works on a simple algorithm and does not require any external capacity to run. The system enables to distribute the traffic congestion evenly throughout the area.

Keywords - traffic, jam factor, HERE Maps API, dynamic traffic control, machine learning

Chapter 1

Introduction

This chapter gives an overview of the project by providing information about the problem definition, existing system and proposed solution.

1.1 Problem Definition

India boasts of being the second-largest road network in the world. The total stretch of the Indian road networks stand at a whopping 5.4 million km! Hence, it forms a huge ultimatum for the Indian Government to provide flawless roads at every step. For any Indian, be it average or millennial, driving through the Indian streets is certainly a hassle that no would like to go through. Some of these mainstream traffic problems include- Poor road quality due to excessive traffic- The extreme congestion of urban roads due to heavily used private vehicles leads to the degradation of the quality of the roads. This leads to incessant traffic problems most of the times. Noise and Air pollution especially in urban areas- The sheer magnitude of traffic problems also gives rise to other health-damaging issues such as the air and sound pollution. So, we propose a solution which controls the traffic dynamically based on various important factors like time of day, climate, condition of roads etc. The system enables to distribute the traffic congestion evenly throughout the area.

1.2 Existing System

The existing systems implement the traffic control system using image processing wherein images of the traffic are captured using cameras and sensors kept at the particular traffic destinations. The issue with the existing systems is:- Complex backgrounds should be eliminated from real photos prior to being inputted for the CNN model. Simple image-processing skills can accommodate this process when adopting real photos in the future.

1.3 Proposed Solution

We have proposed a system that uses an algorithm to control the traffic dynamically based on various important factors like time of day, climate, condition of roads etc. The main aim of our system is to eliminate the use of image processing system for the same and enable the distribute the traffic congestion evenly throughout the area by controlling the traffic dynamically.

1.4 Salient Contribution

Previous research done on traffic management system makes use of image processing and sensors to solve the issue thereby questioning the feasibility of the system. We believe that an algorithm could be used to solve the very issue.

1.5 Organization of the thesis

The Thesis is organized in the following way: Chapter 1 talks about overview of the project selected and defines the motivation behind the project selected and the scope of the project.

Chapter 2 provides us the literature survey done to understand the topic.

Chapter 3 is the Software Project Management Plan (SPMP) of the project. This section includes the software process model selected for the project, tools and techniques that will be used and the project management plan.

Chapter 4 is the Software Requirement Specification which deals with hardware, software, functional and non-functional requirements of the project.

Chapter 5 is the Software Design Document. It deals with the architecture, data needed for the project, user-interface design and a detailed description of the components

Chapter 6 is the Implementation. It gives details about the Technologies, Algorithms used and Pseudo Code if any present related to the algorithm.

Chapter 7 is the Software Testing Document which includes the test plan and the test cases that can be applied to project.

Chapter 8 is about the results generated and the discussion. It describes the work done by the students to overcome the problem and reach the conclusion, it also includes analysis done while designing the architecture for the given problem.

Chapter 9 is about the conclusion and future work. It give details about the conclusion drawn by student after implementing the solutio

Chapter 2

Literature Survey

This chapter discusses about the analysis and research done in the domain of our project and the results that are already published.

2.1 Previous Research

2.1.1 Adaptive Signal Control using Reinforcement Learning

Adaptive signal control system were implemented using deep learning and reinforcement learning algorithm (RL). Instead of a real traffic operation, the present study utilized Vissim, a commercial traffic simulator, as an environment.[1]

A real intersection located in Seoul, Korea was chosen as a test-bed for the simulation. that showed both a real photo and an animation image of the test-bed.It stated that complex backgrounds should be eliminated from real photos prior to being inputted for the CNN model. Simple image-processing skills can accommodate this process when adopting real photos in the future.

The algorithm was trained for 20,000 simulation seconds (= about 5 hours 30 minutes) for each episode. There were 50 episodes simulated for training the model. Thus, the total simulation time was tantamount to 1,000,000 seconds [1]

2.1.2 Smart traffic control system using Image Processing

Smart traffic control system handled traffic using image processing based adaptive signal control-ling. The main aim in designing and developing of the Smart Traffic Signal Simulator is to reduce the waiting time of each lane of the cars and also to maximize the total number of cars that can cross an intersection given the mathematical function to calculate the waiting time. The system worked by detecting the entering objects to the scene, and tracking them throughout the video. The input to the algorithm is the raw video data of a site. This model evinced that the timings for the traffic model did much better, if they were implemented in multiples of five.[2]

2.1.3 Locality constraint distance metric learning for traffic congestion detection

LCDM, a locality constraint distance metric learning is proposed for traffic congestion detection. First of all, an accurate and unified definition of congestion is proposed and the congestion level analysis is treated as a regression problem in the paper. Based on that definition, a dataset consists of 20 different scenes is constructed for the first time since the existing dataset is not diverse for real applications.[3]

To characterize the congestion level in different scenes, the low-level texture feature and kernel regression is utilized to detect traffic congestion level. To reduce the influence among different scenes, a Locality Constraint Distance Metric Learning (LCML) which ensured the local smooth-ness and preserved the correlations between samples is proposed. The extensive experiments con-firm the effectiveness of the proposed method.[3]

Chapter 3

Software Project Management Plan (SPMP)

This Chapter will tell you about the software process model, roles, responsibilities, tools and techniques,tasks, assignments and timetable.

3.1 Introduction

3.1.1 Project Overview

Our project will comprise of a model which controls the traffic dynamically based on various important factors like density of traffic, direction of traffic road length enables to distribute the traffic congestion evenly throughout the area. With the usage of machine learning, we can change the traffic light properties hence providing lesser wait time when required. As the fixed timers system turns the traffic situation into a more chaotic one, our model will predict how to control the traffic at a junction and distribute it evenly to avoid bottlenecks and overly crowded roads.

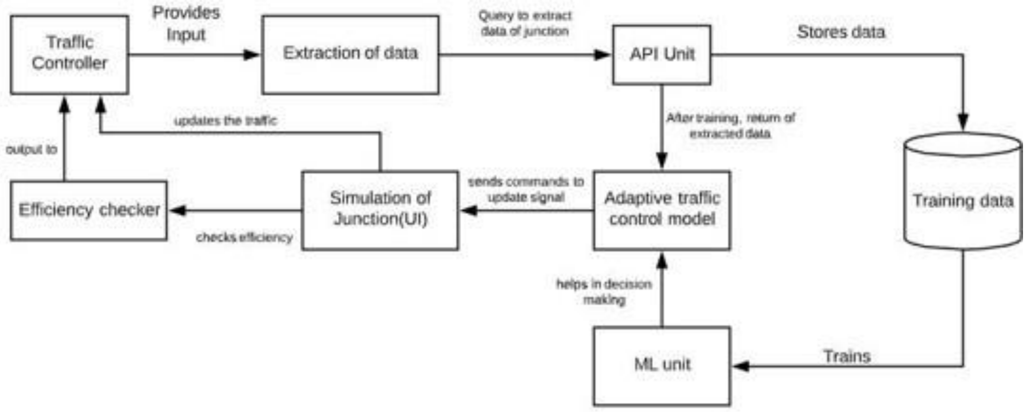


Figure 3.1: Flowchart of working of the application

3.1.2 Project Deliverables

Table 3.1: Project Deliverables

Deliverables	Delivery Date
Proposal and Literature Survey	15/07/2019
Software Requirement Specification (SRS)	10/09/2019
Software Project Management Plan (SPMP)	10/09/2019
Software Design Document (SDD)	05/10/2019
Software Test Document (STD)	14/10/2019
1st module implementation; code, library files	31/10/2019
Updated 1st module along with changes	22/11/2019
2nd module implementation; code, library files	31/01/2020
Updated 2nd module along with changes	15/02/2020
3rd module implementation; code, library files	28/02/2020
Full project implementation for testing	15/03/2020
Project testing and test report	31/03/2020
Project completion along with updates	15/04/2020
Project submission and report	30/04/2020

3.2 Project Organization

3.2.1 Software Process Model

After analyzing, different process models, we accordingly decide to use incremental process model. Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

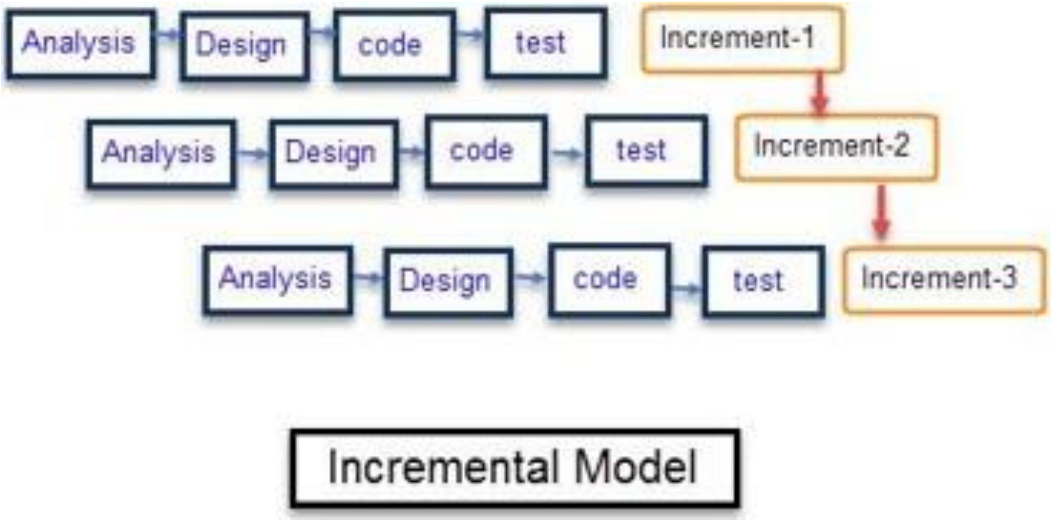


Figure 3.2: Incremental model

3.2.2 Tools and techniques

- Share Latex : Documentation
- Gantt Project : Gantt Chart
- Django framework: Designing UI Frameworks
- Pycharm, Visual studio : Backend Coding
- MySQL: Database
- StarUML: Creation of UML diagrams

3.3 Project Management Plan

3.3.1 Task 1

Analysis and Information gathering

3.3.1.1 Description

Meeting the stakeholders, defining problem statements and gaining requirements which are feasible and realistic, and distinguishing them as functional and non functional requirements.

3.3.1.2 Deliverables and milestones

Formation of problem Statement,Completion of SRS Document

3.3.1.3 Resources Needed

Google Forms for collecting opinions and feasibility reports on requirements, Stakeholders which include the team members of the project viz. project manager, designer, developer and tester and also the project guide.

3.3.1.4 Dependencies and Constraints

If there is no initial discussion with the stakeholders, requirements cannot be specified. All the members stakeholders need to be free at a certain time for certain period for proper discussion.Availability of stakeholders for filling survey form.

3.3.1.5 Risks and Contingencies

Redundancy among the requirements. Formulating the wrong functional and non functional requirements. If there is no initial discussion with team members and stakeholders, problem statement cannot be formulated.

3.3.2 Task 2

Modelling the requirements and design

3.3.2.1 Description

Create the necessary UML based on the requirements gained. Overall understanding of different modules and their integration to form the project.

3.3.2.2 Deliverables and milestones

Completion of UML diagrams

3.3.2.3 Resources Needed

SRS, StarUML

3.3.2.4 Dependencies and Constraints

Depends on the SRS documentation.

3.3.2.5 Risks and Contingencies

Incorrect UML may lead to improper understanding of implementation. Also, the links between different modules are at stake in case of wrong UML diagrams.

3.3.3 Task 3

Designing and implementation of the software project

3.3.3.1 Description

Developing the Prototype, after the prototype is successful initiating the development of the soft-ware product is done.

3.3.3.2 Deliverables and milestones

Prototype, Software Product

3.3.3.3 Resources Needed

Latex Editor,SRS, UML diagrams, SPMP Python

3.3.3.4 Dependencies and Constraints

SRS SPMP: Requirements dependency Task completion constraint

3.3.3.5 Risks and Contingencies

There might be times when the product fails to deliver its purpose. Some requirements that the user wants may be missing. Unsatisfied user is also a factor of risk here.

3.3.4 Task 4

Testing of the Software Product in Real Environment

3.3.4.1 Description

Two tests will be performed:
1:Alpha-testing:Client Side Testing Of Product
2:Beta-testing:Developer Side Testing Of Product

3.3.4.2 Deliverables and milestones

Final Software, User Guide

3.3.4.3 Resources Needed

Personal Computers needed during alpha testing

3.3.4.4 Dependencies and Constraints

Internet Connection needed during Alpha Testing,SRS,SPMP,SDD

3.3.4.5 Risks and Contingencies

There are chances there might be a database failure or network failure hence the software fails to serve its purpose. Delay in fixing defects by development team.

3.3.5 Task 5

Maintenance of Software Product

3.3.5.1 Description

Maintaining the database the software.

3.3.5.2 Deliverables and milestones

Maintenance and Updation

3.3.5.3 Resources Needed

Software product, Python

3.3.5.4 Dependencies and Constraints

Python Language

3.3.5.5 Risks and Contingencies

If proper maintenance plan is not made quality of product will deteriorate

Chapter 4

Software Requirement Specification (SRS)

This chapter discusses about the description of a software system to be developed. It lays out functional and non-functional requirements, and also a set of use cases that describe user interactions that the software must provide.

4.1 Introduction

4.1.1 Product Overview

In order to deal with the ever increasing traffic congestion in the country, our proposed system will make use of a simple algorithm based on machine learning. The system will consist of a portal which will accept inputs as road name and display its equivalent coordinates from the API. This data will be real-time and dynamic. Furthermore, our system will dynamically divide the roads according to their jam factors based on the amount of traffic i.e from highest to lowest on each road. Accordingly, the system will help distribute the traffic evenly throughout.

4.2 Specific Requirements

4.2.1 External Interface Requirements

Since, it is a web based application HTTP or HTTPS protocol is followed.

4.2.1.1 User Interface

The Adaptive Traffic Control System screen displays shall conform to the Process Impact Internet Application User Interface Standard, Version 2.0. The Web pages shall permit complete navigation and data selection and display using the keyboard alone, in addition to using mouse and keyboard combinations.

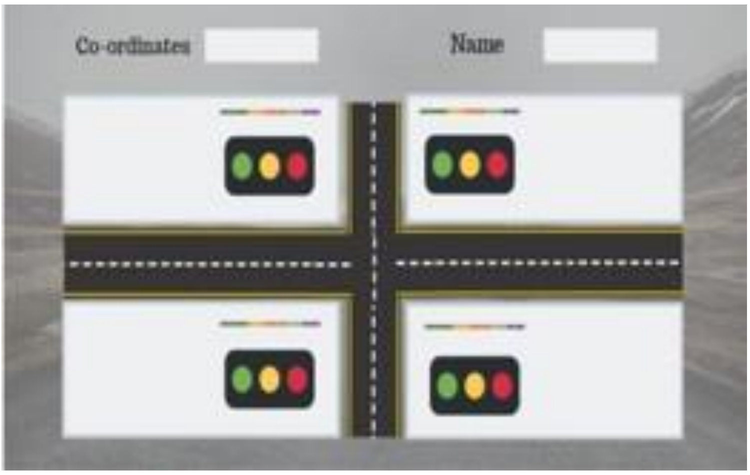


Figure 4.1: User interface

4.2.1.2 Hardware Interfaces

Since the proposed product is a web application with no designated hardware, there is no direct hardware interface for the system.

4.2.1.3 Software Interfaces

The following are the software interfaces used for application:

- MySQL : MySQL is an open-source relational database management system.
- PyCharm : PyCharm is an integrated development environment used in computer program-ming, specifically for the Python language and used for Django development.
- Django : Django is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern
- Scikit-learn : Scikit-learn is a free software machine learning library for the Python program-ming language.DESC: The system shall allow registered users to enter into the system when they have forgotten the password. The system then should ask for a new password to be set by user.

4.2.1.4 Communications Interfaces

This system will follow basic protocols under HTTP and HTTPS.

4.2.2 Software Product Features

- Extract the details from the website for the given co-ordinates of the junction.
- Simulate the junction by using Machine Learning Algorithm and switch the signals accordingly.
- Distribute the traffic to the surrounding lanes if the traffic jam factor is more than the thresh-old value.
- Control the traffic signal timer dynamically based on the congestion.

4.2.3 Software System Attributes

4.2.3.1 Reliability

Application will be always 100 percent functional in each and every situation.

4.2.3.2 Availability

Regular backups of the database will be made to ensure consistency of data if the database crashes.

4.2.3.3 Security

Proper authorizations will be provided to people administering databases and no unauthorized people will be able to access code or databases

4.2.3.4 Maintainability

The code will be written in a way that any further features can be added easily.

4.2.3.5 Portability

Complete website will be made on languages like HTML, CSS, python etc. which are supported on all modern operating systems.

4.2.4 Database requirements

The inbuilt database of Django i.e. sqlite will be used for registration. Username is the unique key for the same. MySQL database will be used. The database will be stored with unique ID for each entry for statistics and efficiency factor.

Chapter 5

Software Design Document (SDD)

This chapter discusses about the design and architecture used for our system.Also it provides information of components used in the system.

5.1 Introduction

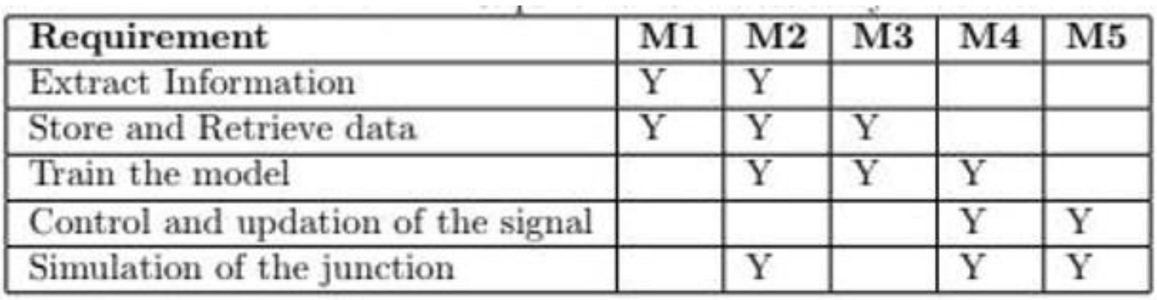
Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of the application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once the system requirement have been specified and analyzed, system design is the first of the three technical activities - design, code and test is to build and verify software. [4]
The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. [4]

5.1.1 Design Overview

Our project design will comprise of a portal which will accept inputs as road name and display its equivalent coordinates from the API. There will be four different roads on the main dashboard. Each of these roads will include the name of the road, a timer and traffic lights for simulation. The data displayed will be real-time and dynamic. Furthermore, our system will dynamically divide the roads according to their jam factors based on the amount of traffic i.e from highest to lowest on each road. Accordingly, the system will help distribute the traffic evenly throughout.

5.2 Requirement Matrix

The table shows which system components satisfies each of the functional requirements from the Software Requirement Specification Document.



Requirement	M1	M2	M3	M4	M5
Extract Information	Y	Y			
Store and Retrieve data	Y	Y	Y		
Train the model		Y	Y	Y	
Control and updation of the signal				Y	Y
Simulation of the junction		Y		Y	Y

Figure 5.1: Requirement Matrix

5.3 System architectural Design

5.3.1 Chosen System Architecture

Our project follows the Data-centered architecture. In this architecture, we have a team of admins who manage the traffic database and distribute the traffic accordingly by taking into consideration, the traffic data and its various factors.

5.3.2 Alternate System Architecture

5.3.2.1 Data-Centric Architecture

We are not following the Data-centric Architecture as our project only provides simple interface to generate music from sheet music using pre-trained weighted model. We do not store any data given by the user in the system.

5.3.2.2 Data flow architecture

We are not following the Data flow architecture since data flow is a series of events and our project revolves around central admin.

5.3.2.3 Call and Return architecture

We could have chosen the call and return architecture as the program structure is easy to modify but the data-centered architecture simply offers more flexibility and caters to all the needs of our project.

5.3.2.4 Layered architecture

Layered structure proved to be more complicated hence it was much more preferable to avoid that.

5.3.2.5 Object Oriented architecture

Since object-oriented architecture is the latest version of call and return architecture, the problems arising in the same are similar to that of call and return architecture.

5.4 Data Design

This section explains how the information domain of our system is transformed into data structures. It also describes how the major data or system entities are stored, processed and organized.

5.4.1 Data Flow Specifications

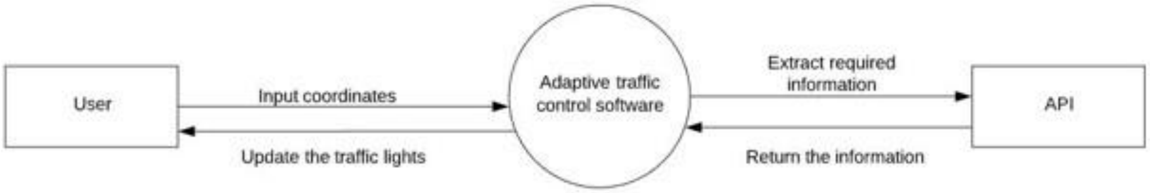

User
Input coordinates
Update the traffic lights
Adaptive traffic control software
Extract required information
Return the information
API

Figure 5.2: DFD Level 0

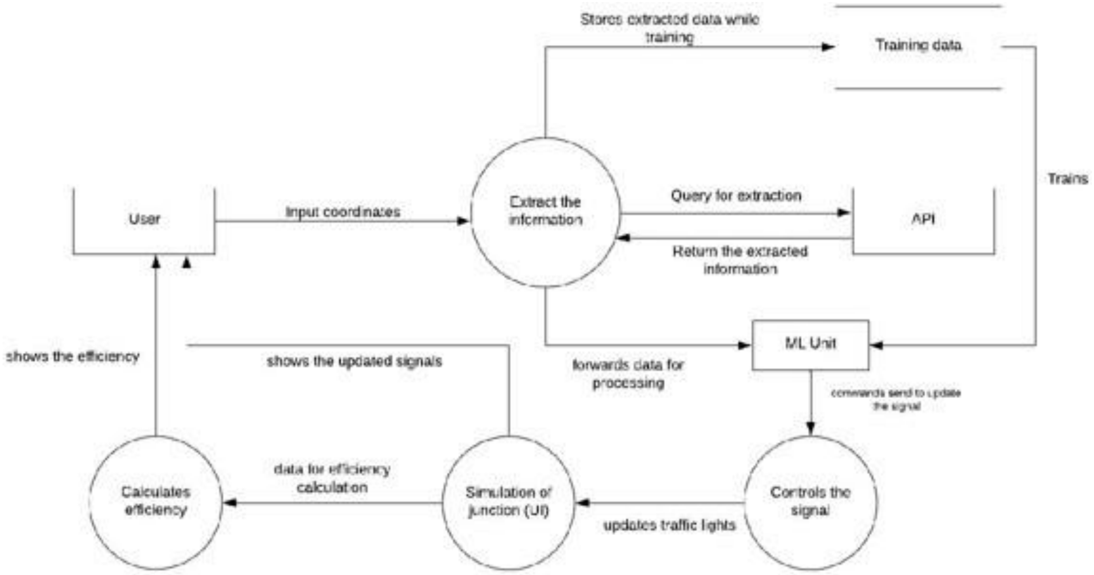

Stores extracted data while training
Training data
Trains
User
Input coordinates
Extract the information
Query for extraction
API
Return the extracted information
ML Unit
forwards data for processing
commands send to update the signal
shows the efficiency
shows the updated signals
Calculates efficiency
data for efficiency calculation
Simulation of junction (UI)
updates traffic lights
Controls the signal

Figure 5.3: DFD Level 1

5.5 System Interface Description

5.5.1 System Architecture

Table 5.1: Use case 1

Use Case ID:	1		
Use Case Name:	Receiving data from api		
Created by:		Date created:	October 3,2019
Actors	System		
Description:	System sends an API call to here maps api and then system receives data from HERE maps API		
Preconditions:	None		
Postconditions:	System should successfully receive accurate data without any delay		
Trigger:	none		

Table 5.1 – *Continued on next page*

Table 5.1 – *Continued from previous page*

Normal flow:	System sends an API call to here maps api and then system receives data from HERE maps API
Alternative flows:	None
Exceptions:	Server is down , database connectivity

Table 5.2: Use case 2

Use Case ID:	2		
Use Case Name:	Prediction of Traffic		
Created by:		Date created:	October 3,2019
Actors	System		
Description:	System processes the data received from api through trained model and predicts the output		
Preconditions:	Data received from API should be accurate		
Postconditions: Trigger:	System should successfully predict the traffic with the help of trained model none		
Normal flow:	System processes the data received from API through trained model and predicts the output		
Alternative flows:	None		
Exceptions:	SSystem is down, No power supply , database connectivity		

Table 5.3: Use case 3

Use Case ID:	3		
Use Case Name:	Processing of output		
Created by:		Date created:	October 3,2019
Actors	System		
Description:	System gives the output based on the density of traffic predicted		
Preconditions:	System must correctly predict the traffic in the region		
Postconditions: Trigger:	System should successfully give the output to decrease the traffic in the region none		
Normal flow:	SSystem gives the output based on the density of traffic predicted		
Alternative flows:	None		
Exceptions:	System is down, No power supply , database connectivity		

5.6 Detailed description of components

5.6.1 API Unit

The Application Programming Interface will be used to extract the signal and traffic information.This data will form the basis of the project for further processing.

5.6.2 Adaptive traffic control system

The system that we use will be the one to request the necessary information from the API in order for the data to be further processed and acted upon. It will also help in training the data creating classifier for the project

5.6.3 Database

The information retrieved from the API will be stored in the database in order to be utilized for training purpose.

5.6.4 ML Processing Unit

The ML Processing Unit will be used to create classifiers and train the acquired data,and use the trained data to output important information with respect to the objective of the project.

5.6.5 Efficiency calculator

Efficiency calculator will calculate the efficiency of our system with respect to the real life traffic decisions, hence calculating the correctness of the software.

5.7 User Interface Design

5.7.1 Description of User Interface

5.7.1.1 Screen Images

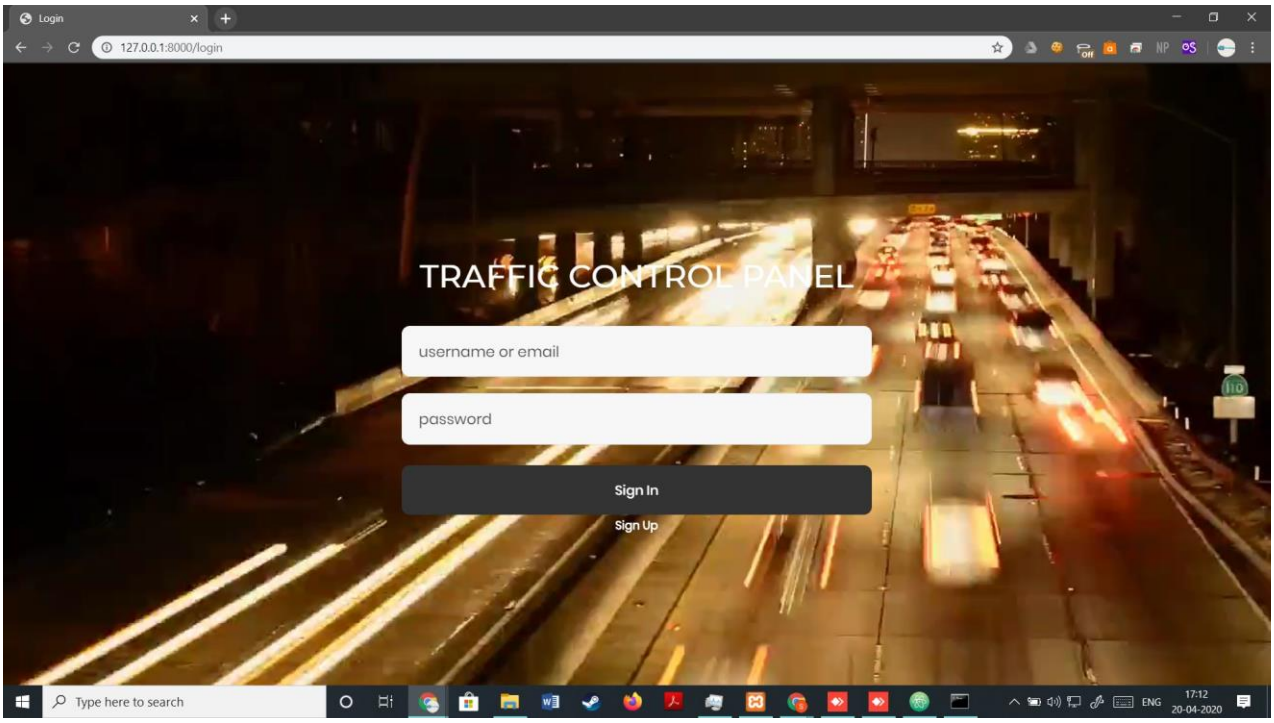
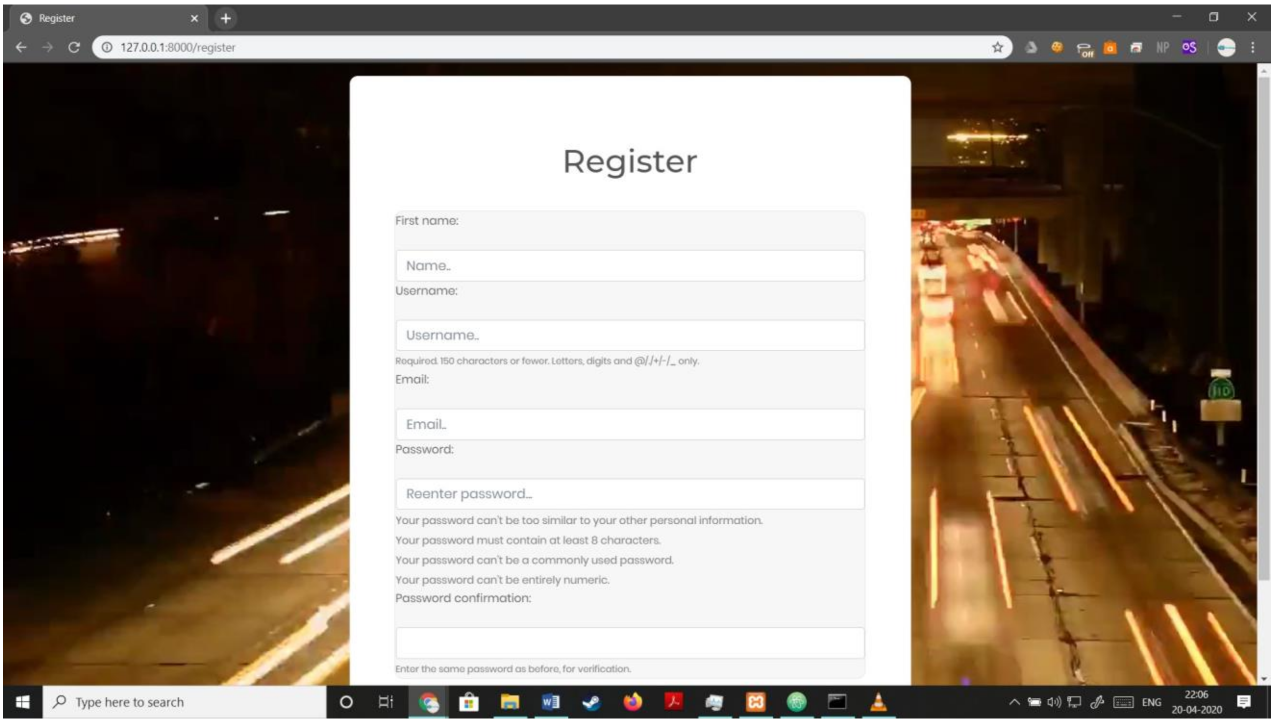


Figure 5.4: Login page



Register

First name:

Name..

Username:

Username..

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

Email..

Password:

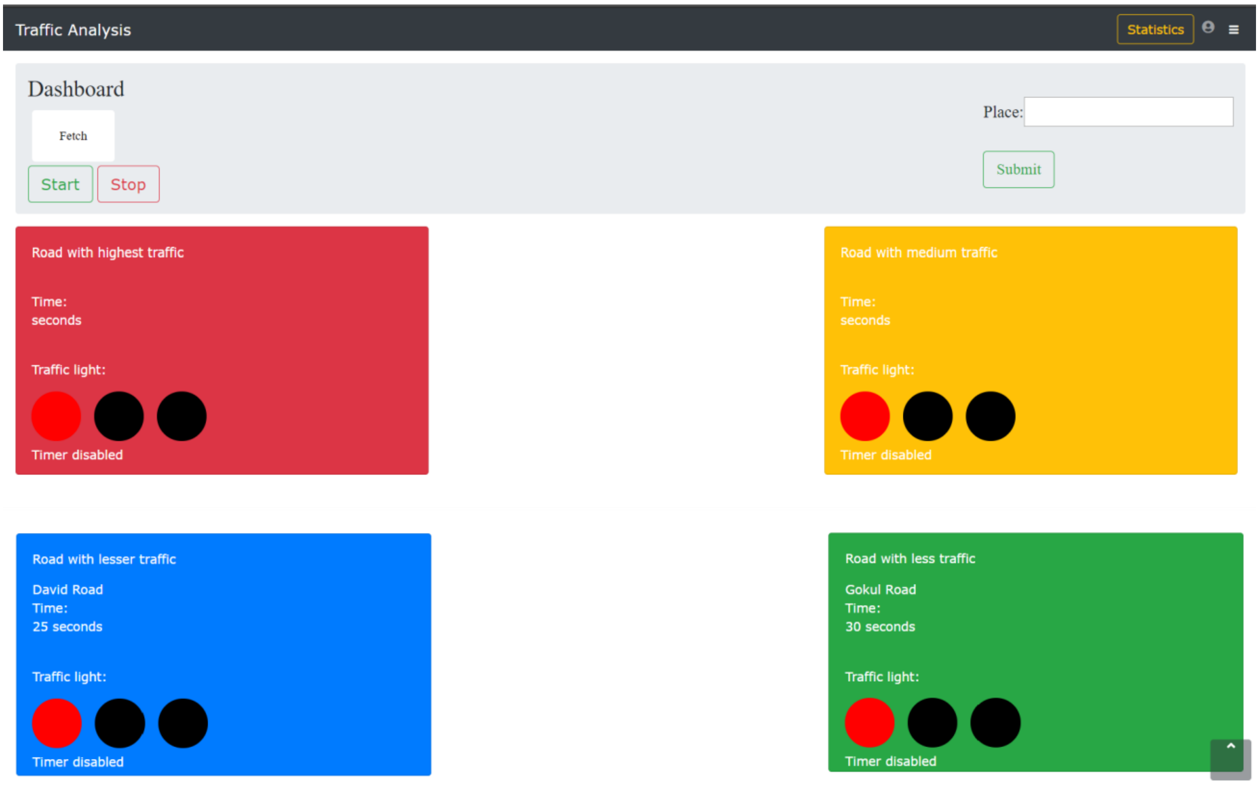
Reenter password...

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Figure 5.5: Register page



Dashboard

Place:

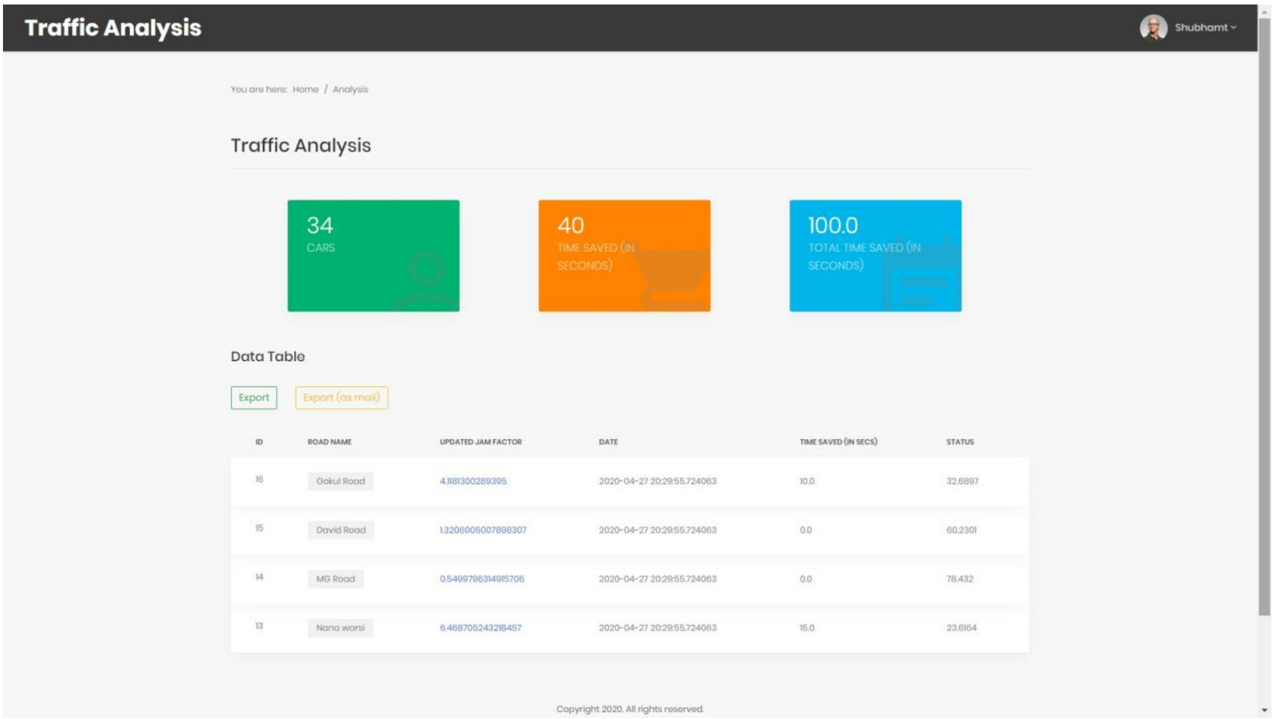
Road with highest traffic
Time:
seconds
Traffic light:
Timer disabled

Road with medium traffic
Time:
seconds
Traffic light:
Timer disabled

Road with lesser traffic
David Road
Time:
25 seconds
Traffic light:
Timer disabled

Road with less traffic
Gokul Road
Time:
30 seconds
Traffic light:
Timer disabled

Figure 5.6: Analysis page



You are here: Home / Analysis

Traffic Analysis

34
CARS

40
TIME SAVED (IN SECONDS)

100.0
TOTAL TIME SAVED (IN SECONDS)

Data Table

ID	ROAD NAME	UPDATED JAM FACTOR	DATE	TIME SAVED (IN SECS)	STATUS
16	Gokul Road	4.1181300289395	2020-04-27 20:29:55.724063	10.0	32.6897
15	David Road	1.3206005007898307	2020-04-27 20:29:55.724063	0.0	60.2301
14	MG Road	0.5499796314915706	2020-04-27 20:29:55.724063	0.0	78.432
13	Nana worsi	6.468705243218457	2020-04-27 20:29:55.724063	15.0	23.6164

Copyright 2020. All rights reserved.

Figure 5.7: Statistics and Efficiency factor page

5.7.1.2 Objects and actions

Login page

- Email text field: This components accepts email address as input.This is the sole identification by which the user can avail our services.
- Password text field: This component takes password as input. If password doesn't match, it doesn't allow the user to access the portal.
- Sign in button: Submits the data for authentication and lets the user to the next page if authenticated.
- Sign up: This button redirects to register page

Register page

- First Name text field: This field takes name as the input.
- Username text field: This component takes username as input. The user name should not be similar to first name.
- Email text field: This components accepts email address as input.This is the sole identification by which the user can avail our services.
- Password text field: This component takes password as input. The given conditions on the page should be satisfied to set the password.
- Confirm password text field: This component takes the password again to recheck the same.
- Submit button: Submits the data if all the fields are filled and stores to the database and redirects to login page
- Sign in: This button redirects to login page

Analysis

- Place text field: This component accepts the name of the road as input.
- Submit button: This component returns the co ordinates of the place entered in the place text field.

- Fetch button: This component fetches the data from the API for the place.

- Start button: This component starts the simulation for all the given roads of the place.

- Stop button: This button when clicked, leads to stopping of the simulation.

Statistics and Efficiency factor page

- Export button: This component exports the page data into excel format for the user.

- Export to mail: This component mails the page data in Excel format to the user’s email

Chapter 6

Implementation

This chapter provides technologies, Algorithms used and the components used in the project.

6.1 Technologies used

- **Developing Environment**

  Processors: Any above 2.4GHz.

  Ram: 8GB (or more).

  Input Devices: Keyboard and Mouse.

  Software Requirements

- **Operating System**

  Windows, Ubuntu: Our application will run on both these operating system as they are the most widely used and available.

- **Platform**

  Web Browsers: Any standard web browsers such as Google Chrome, Internet Explorer, Mozilla Firefox etc.

- **Database**

  MySQL: As it the easy to use and compatible with both Windows and Ubuntu.

- **Client Side**

  Django: For UI development we used Django as it is quite light weight

- **Server Side**

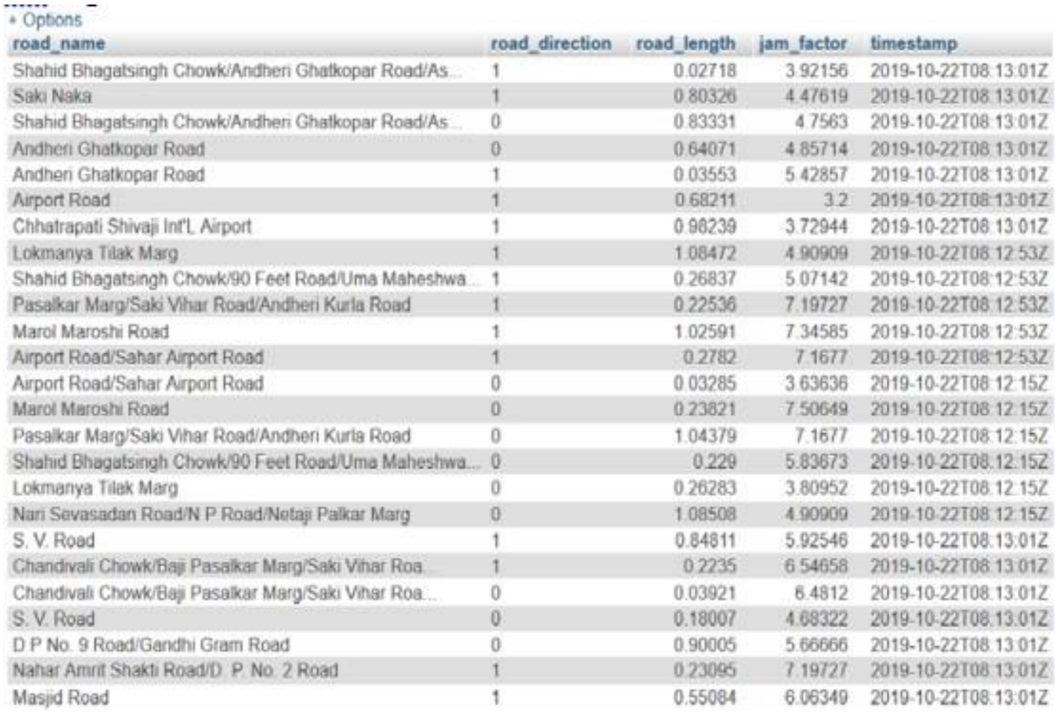
  Python: It is an interpreted, high-level, general-purpose programming language.

  JavaScript: It is a high-level, dynamic, weakly typed, object-based, multi-paradigm, and interpreted programming language

6.2 Algorithm

We will be using **Naive Bayes algorithm** for our system. To implement our goals and objectives we first extract data from HERE Maps API[4], which provides us with the traffic density data in the form of JSON or XML format. We get various features regarding the traffic on the road, such as the average speed of any car on that particular road, the length of the road, the maximum speed of any vehicle on the road and the jam factor (the traffic density) on that stretch of road.

Our initial step was to setup our environment for data collection from the API and storing it into the database. During the collection of data, we pre-processed the data storing only the attributes that were required for further implementation of our model. We even performed cleaning of the data to get rid of inconsistent data such as zero values.

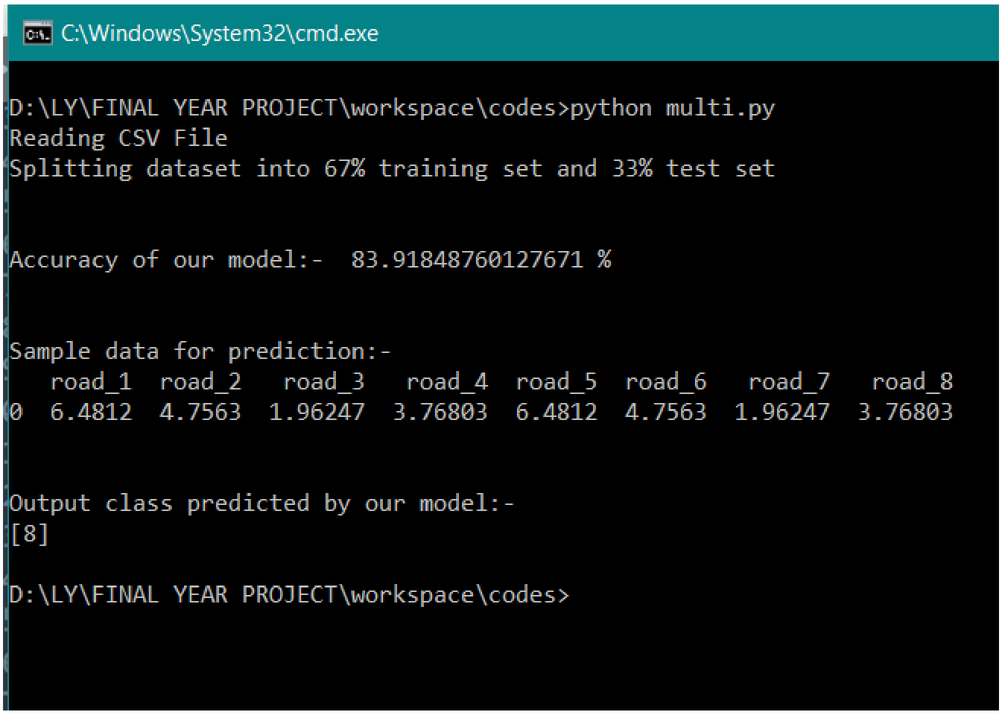


+ Options

road_name	road_direction	road_length	jam_factor	timestamp
Shahid Bhagatsingh Chowk/Andheri Ghatkopar Road/As...	1	0.02718	3.92156	2019-10-22T08:13:01Z
Saki Naka	1	0.80326	4.47619	2019-10-22T08:13:01Z
Shahid Bhagatsingh Chowk/Andheri Ghatkopar Road/As...	0	0.83331	4.7563	2019-10-22T08:13:01Z
Andheri Ghatkopar Road	0	0.64071	4.85714	2019-10-22T08:13:01Z
Andheri Ghatkopar Road	1	0.03553	5.42857	2019-10-22T08:13:01Z
Airport Road	1	0.68211	3.2	2019-10-22T08:13:01Z
Chhatrapati Shivaji Int'L Airport	1	0.98239	3.72944	2019-10-22T08:13:01Z
Lokmanya Tilak Marg	1	1.08472	4.90909	2019-10-22T08:12:53Z
Shahid Bhagatsingh Chowk/90 Feet Road/Uma Maheshwa...	1	0.26837	5.07142	2019-10-22T08:12:53Z
Pasalkar Marg/Saki Vihar Road/Andheri Kurla Road	1	0.22536	7.19727	2019-10-22T08:12:53Z
Marol Maroshi Road	1	1.02591	7.34585	2019-10-22T08:12:53Z
Airport Road/Sahar Airport Road	1	0.2782	7.1677	2019-10-22T08:12:53Z
Airport Road/Sahar Airport Road	0	0.03285	3.63636	2019-10-22T08:12:15Z
Marol Maroshi Road	0	0.23821	7.50649	2019-10-22T08:12:15Z
Pasalkar Marg/Saki Vihar Road/Andheri Kurla Road	0	1.04379	7.1677	2019-10-22T08:12:15Z
Shahid Bhagatsingh Chowk/90 Feet Road/Uma Maheshwa...	0	0.229	5.83673	2019-10-22T08:12:15Z
Lokmanya Tilak Marg	0	0.26283	3.80952	2019-10-22T08:12:15Z
Nari Sevasadan Road/N P Road/Netaji Palkar Marg	0	1.08508	4.90909	2019-10-22T08:12:15Z
S. V. Road	1	0.84811	5.92546	2019-10-22T08:13:01Z
Chandivali Chowk/Baji Pasalkar Marg/Saki Vihar Roa...	1	0.2235	6.54658	2019-10-22T08:13:01Z
Chandivali Chowk/Baji Pasalkar Marg/Saki Vihar Roa...	0	0.03921	6.4812	2019-10-22T08:13:01Z
S. V. Road	0	0.18007	4.68322	2019-10-22T08:13:01Z
D P No. 9 Road/Gandhi Gram Road	0	0.90005	5.66666	2019-10-22T08:13:01Z
Nahar Amrit Shakti Road/D. P. No. 2 Road	1	0.23095	7.19727	2019-10-22T08:13:01Z
Masjid Road	1	0.55084	6.06349	2019-10-22T08:13:01Z

Figure 6.1: Database

For further implementation for input to our machine learning unit, we preprocessed the data according to the requirements of our project. The data was fed into the Machine Learning Unit, in which we used sci-kit learn libraries for training our model and saving the same model which will be used later during the simulation of traffic. Output of our trained model (Accuracy of our model):-



```
D:\LY\FINAL YEAR PROJECT\workspace\codes>python multi.py
Reading CSV File
Splitting dataset into 67% training set and 33% test set


Accuracy of our model:-  83.91848760127671 %


Sample data for prediction:-
   road_1  road_2   road_3   road_4  road_5  road_6   road_7   road_8
0  6.4812  4.7563  1.96247  3.76803  6.4812  4.7563  1.96247  3.76803


Output class predicted by our model:-
[8]

D:\LY\FINAL YEAR PROJECT\workspace\codes>
```

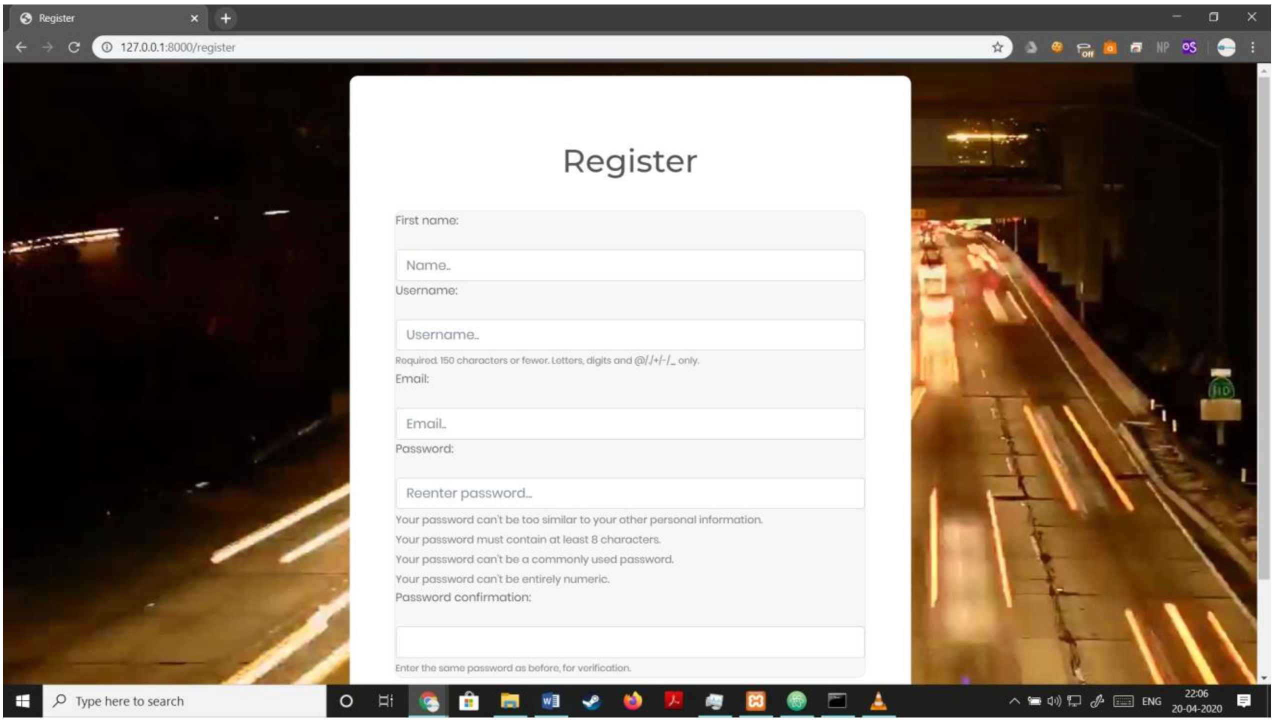
Figure 6.2: Efficiency

There are different situations that are taken into consideration when we get the output from our ML unit. Since we have considered a junction of 4 roads and it’s subsequent connecting roads, we are considering multiple possibilities that we can have in our simulation model. These multiple possibilities include, but are not limited to, the direction of traffic flow when there is a green light, ie, left, right, forward.

Our main aim is to control the traffic signals dynamically based on the current traffic density and considering the historic traffic density values. Among the roads that we have considered, the road having the highest jam factor (traffic density) will be considered first. Time for a particular action, ie, a red light or a green light will be given based on various factors, mainly the jam factor.

To measure the efficiency of our system, whether or not we have managed to reduce the traffic congestion on the roads using our simulated model, we will compare our simulated values with the real time values obtained from the API. We will measure the difference between these values, and this difference when expressed in terms of percentage will define how better traffic can be managed/ controlled using our implemented model.

6.3 Implementation of user interface



Register

First name:

Name..

Username:

Username..

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

Email..

Password:

Reenter password...

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Figure 6.3: Register

Registration page- If the user has not registered, he/she can do so by going to the account registration page and entering details such as:- first name of the user, username, email id and password. User must then enter the same password in the password confirmation box. After entering these details, the user must click the submit button in order to register his/her account. After registering, the user can then login to the portal by using the registered email id and password.

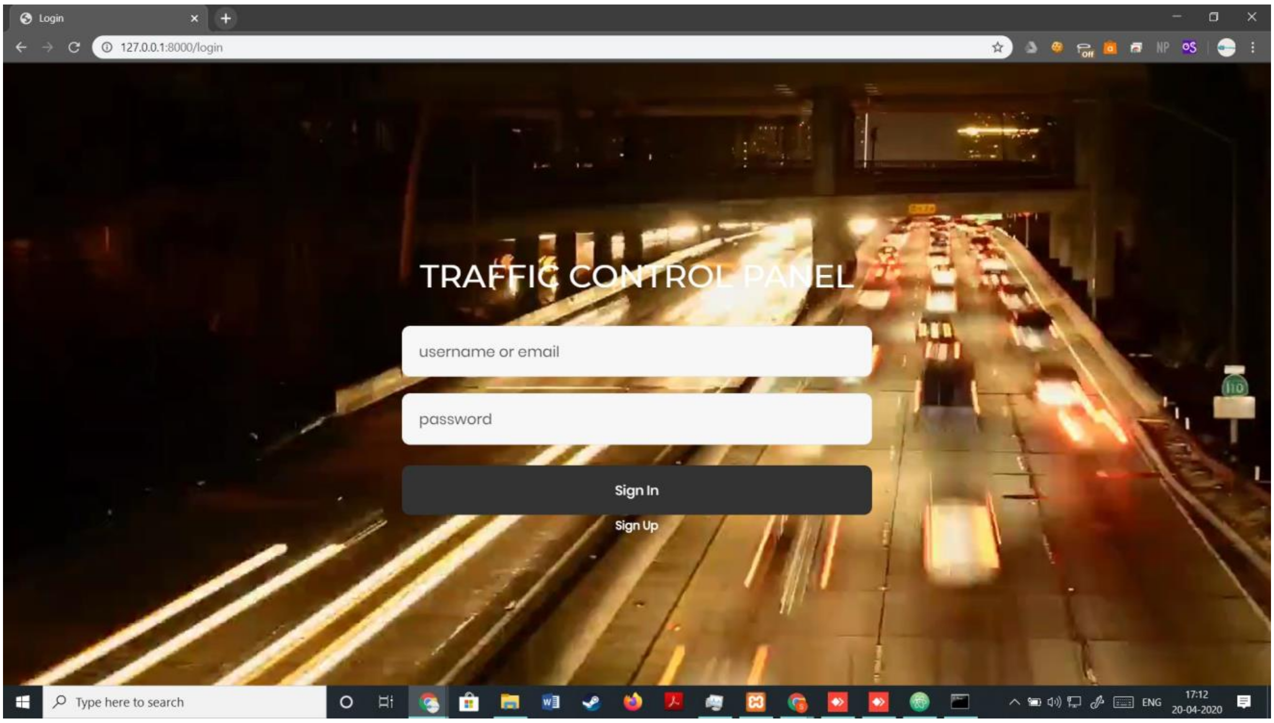
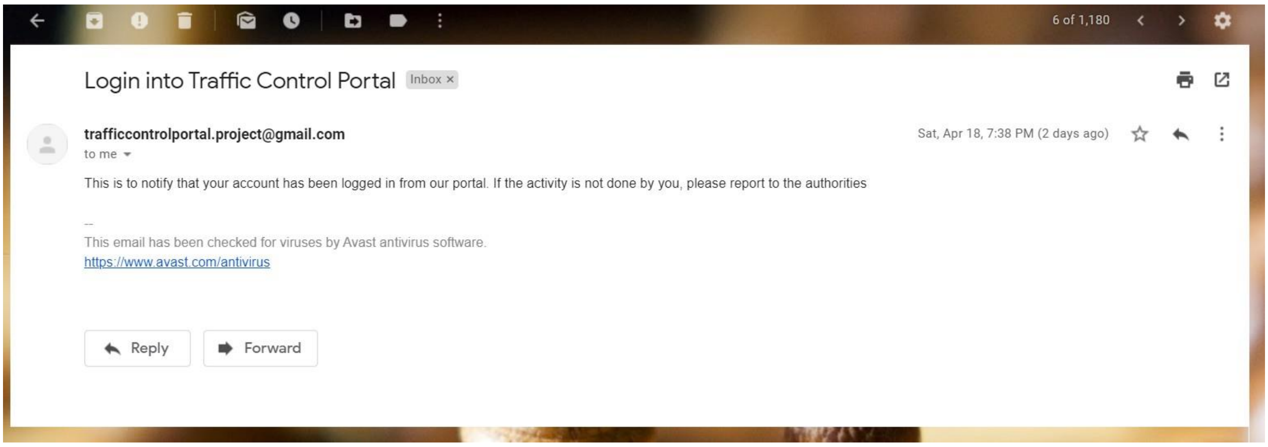


Figure 6.4: Login

Administrator sign in page-Administrator can sign in using his/her email id and password provided to him/her. As soon as he/she logs in,email will be sent to the registered mail indicating the activity done. If someone tries to log in with the wrong password, an email will be sent to notify the user about the unusual activity.



Login into Traffic Control Portal

trafficcontrolportal.project@gmail.com
to me

Sat, Apr 18, 7:38 PM (2 days ago)

This is to notify that your account has been logged in from our portal. If the activity is not done by you, please report to the authorities

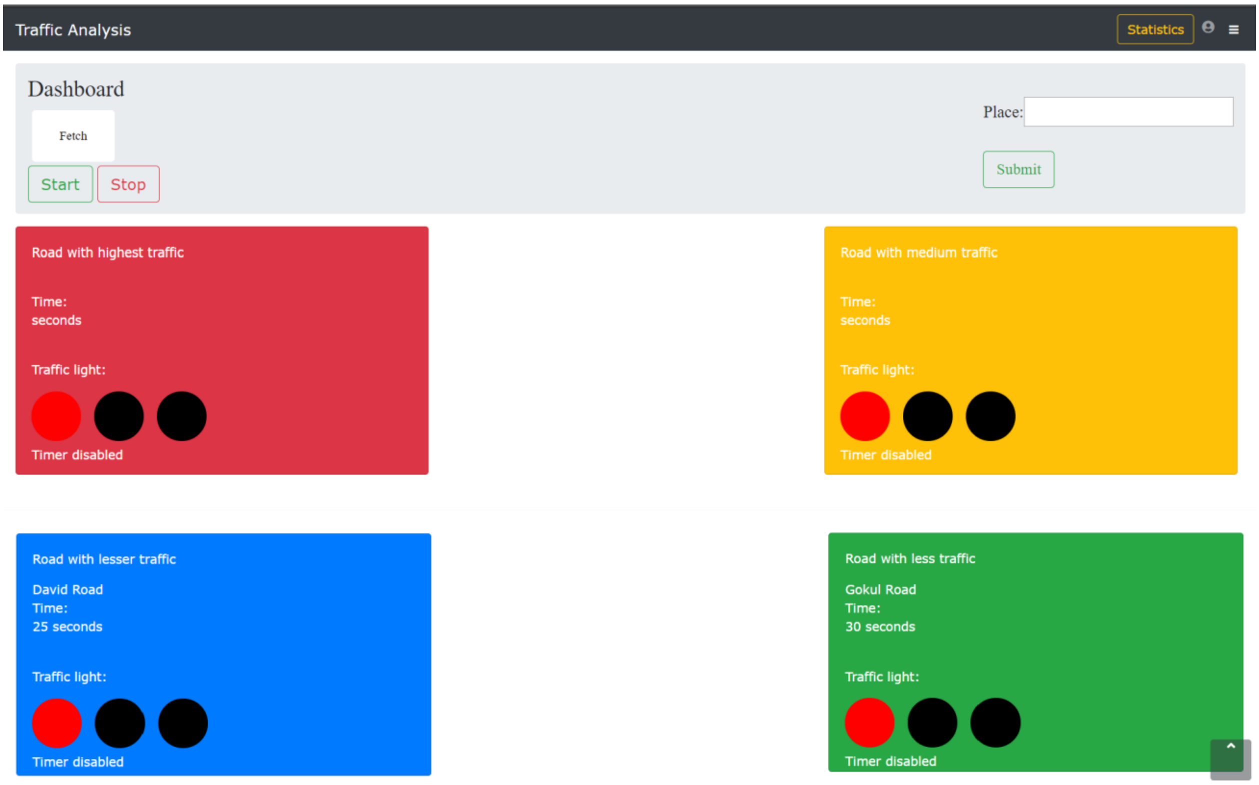
--
This email has been checked for viruses by Avast antivirus software.
https://www.avast.com/antivirus

Figure 6.5: Login activity mail notification

After logging in, you will be directed to this dashboard page. It contains 4 boxes for the 4 roads at the junction. Each box contains:

- Name of the road
- Timer
- Traffic light

According to the jam factor, the road will be divided by the amount of traffic on each road i.e. highest traffic, medium traffic, less traffic, lesser traffic.



Road with highest traffic

Time:
seconds

Traffic light:

Timer disabled

Road with medium traffic

Time:
seconds

Traffic light:

Timer disabled

Road with lesser traffic

David Road
Time:
25 seconds

Traffic light:

Timer disabled

Road with less traffic

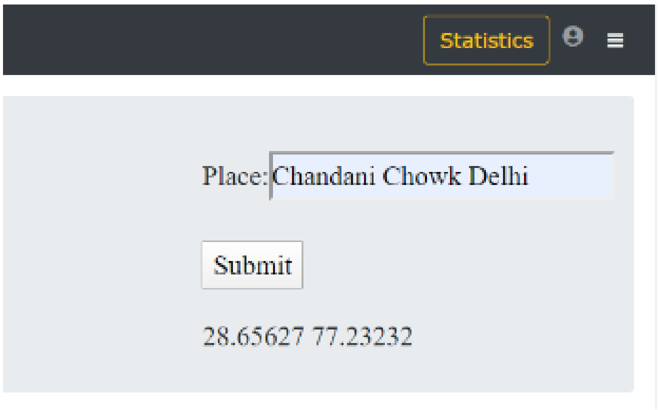
Gokul Road
Time:
30 seconds

Traffic light:

Timer disabled

Figure 6.6: Analysis Dashboard

Place- This takes the road junction as input and returns the coordinates of the same.

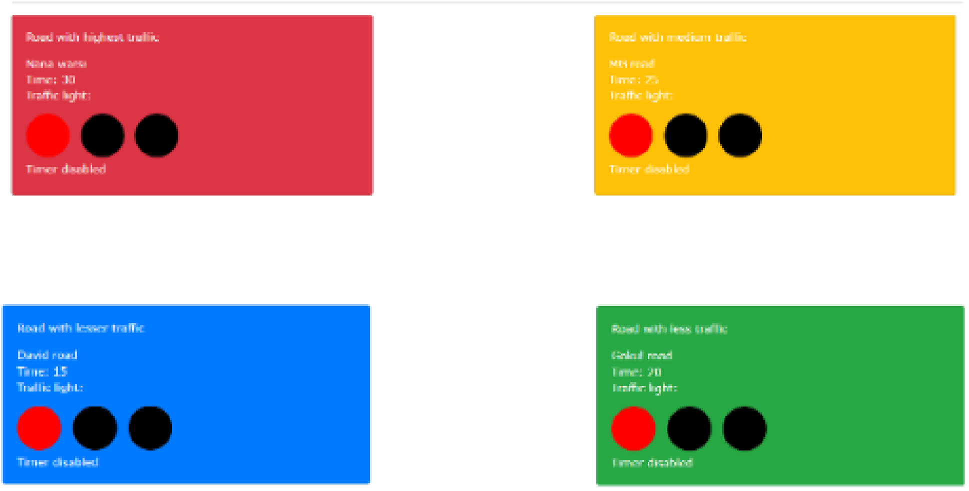


Place: Chandani Chowk Delhi

28.65627 77.23232

Figure 6.7: Coordinates fetched

Fetch: After clicking this, API for the real time data of the given coordinates will be called. It returns real time data.



Road with highest traffic

Nana warsi
Time: 30
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with medium traffic

MG road
Time: 25
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with lesser traffic

David road
Time: 15
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with less traffic

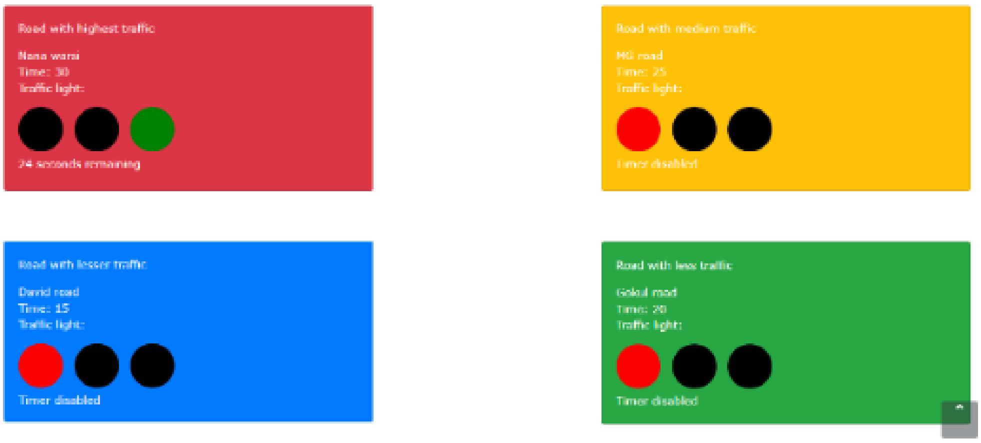
Gokul road
Time: 20
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Figure 6.8: Fetch button clicked

Clicking on start will start the simulation of the system. After clicking, it gets disabled until the simulation is completed.



Road with highest traffic

Nana warsi
Time: 30
Traffic light:

⚫ ⚫ 🟢

24 seconds remaining

Road with medium traffic

MG road
Time: 25
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with lesser traffic

David road
Time: 15
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with less traffic

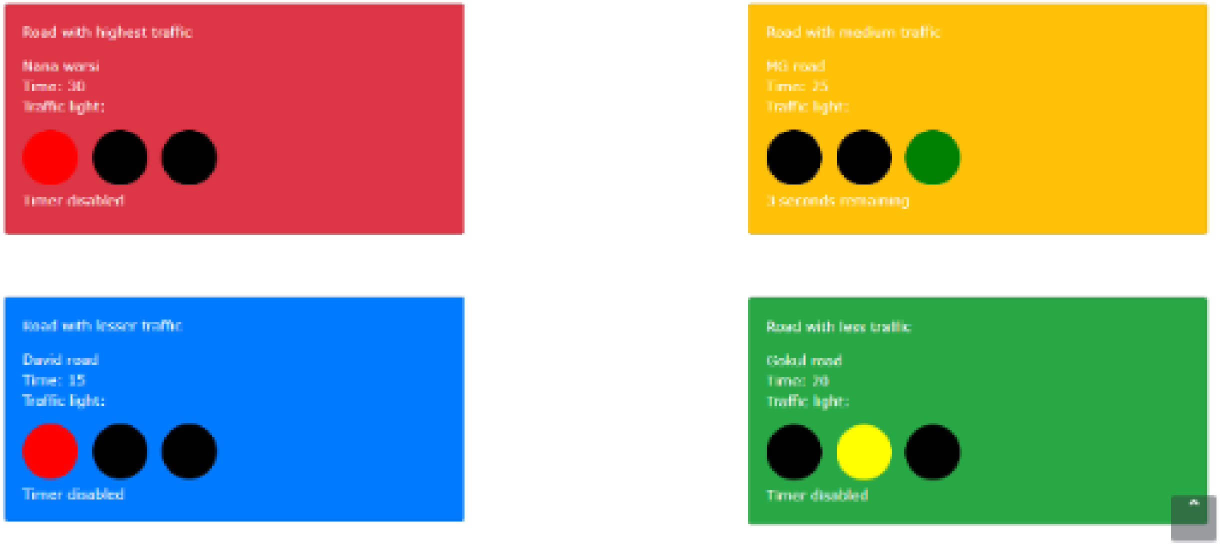
Gokul road
Time: 20
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Figure 6.9: Simulation started

When there are 3 seconds left, the next road’s signal gets yellow.



Road with highest traffic

Nana warsi
Time: 30
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with medium traffic

MG road
Time: 25
Traffic light:

⚫ ⚫ 🟢

3 seconds remaining

Road with lesser traffic

David road
Time: 15
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with less traffic

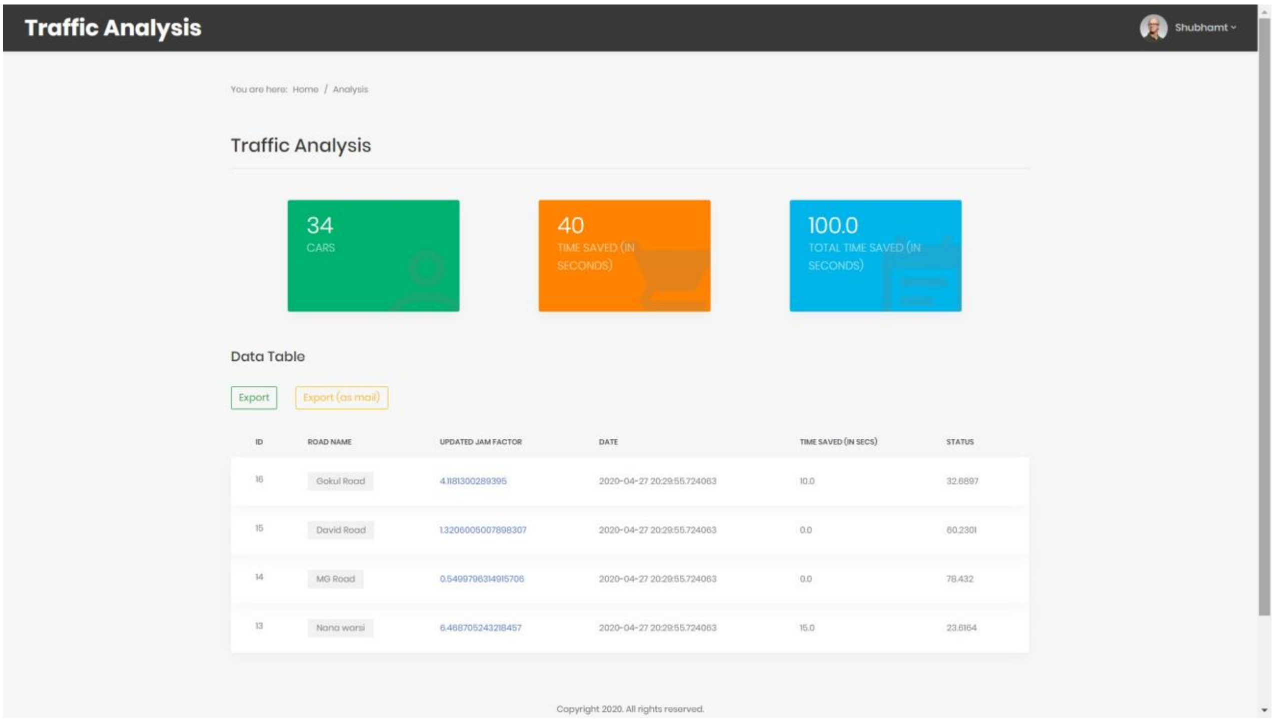
Gokul road
Time: 20
Traffic light:

⚫ 🟡 ⚫

Timer disabled

Figure 6.10: Working of system

The simulation continues without any action on page as Flask API auto fetches the data from traffic API without reloading the page.When end button is clicked, the simulation stops.



Traffic Analysis

34 CARS

40 TIME SAVED (IN SECONDS)

100.0 TOTAL TIME SAVED (IN SECONDS)

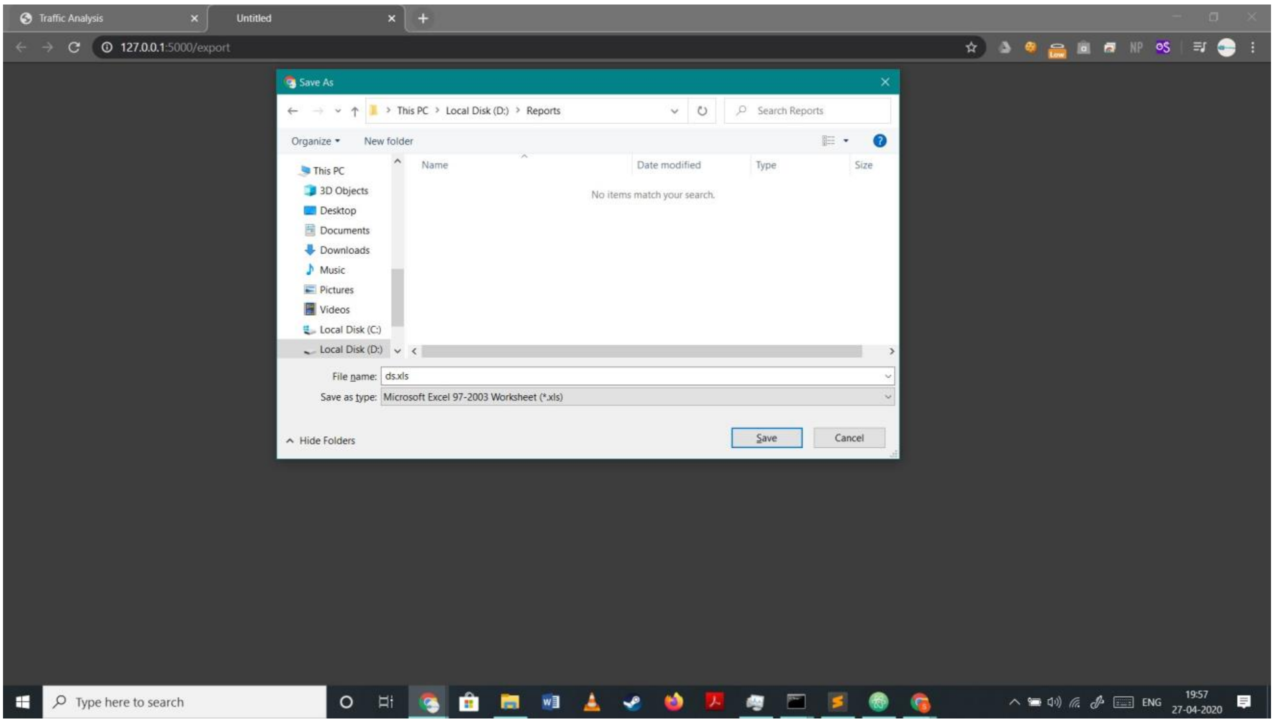
Data Table

ID	ROAD NAME	UPDATED JAM FACTOR	DATE	TIME SAVED (IN SECS)	STATUS
16	Gokul Road	4.1181300289395	2020-04-27 20:29:55.724063	10.0	32.6897
15	David Road	1.3206005007898307	2020-04-27 20:29:55.724063	0.0	60.2301
14	MG Road	0.5499796314915706	2020-04-27 20:29:55.724063	0.0	78.432
13	Nana warsi	6.468705243218457	2020-04-27 20:29:55.724063	15.0	23.6164

Copyright 2020. All rights reserved.

Figure 6.11: Statistics and Efficiency factor page

The statistics button leads to the efficiency and statistics calculated page. The boxes above show the total statistics of total cars left from the area using our model,time saved in last cycle total time saved from the time system was started. The cars box shows us the total number of cars that left the road and didn't get stuck in traffic because of our system. The time saved depicts that how much the time was saved in last cycle whereas the total time saved shows time saved in total by our system.When clicked on export, the data is exported in Excel sheet format.



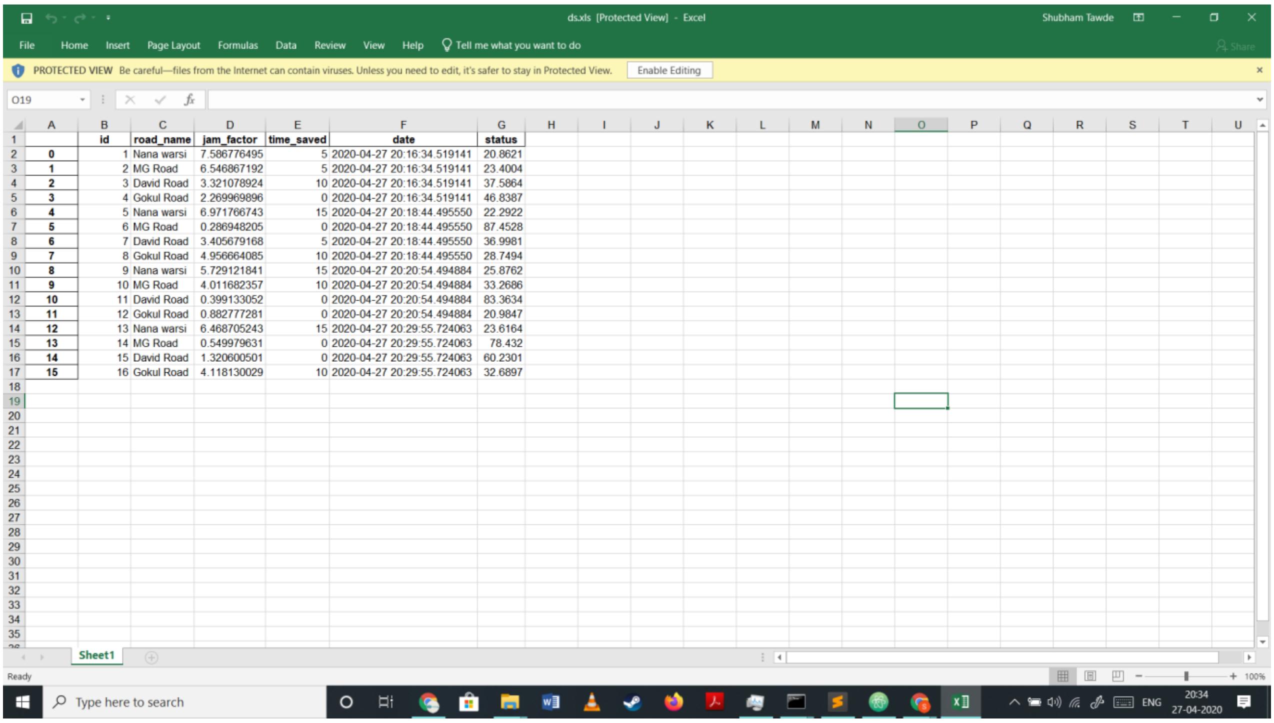
No items match your search.

File name: ds.xls

Save as type: Microsoft Excel 97-2003 Worksheet (*.xls)

Figure 6.12: Export button clicked

The export button gives you the option to save the data to an excel sheet. The excel sheet contains the road name, jam factor, time saved, timestamp, efficiency percentage.

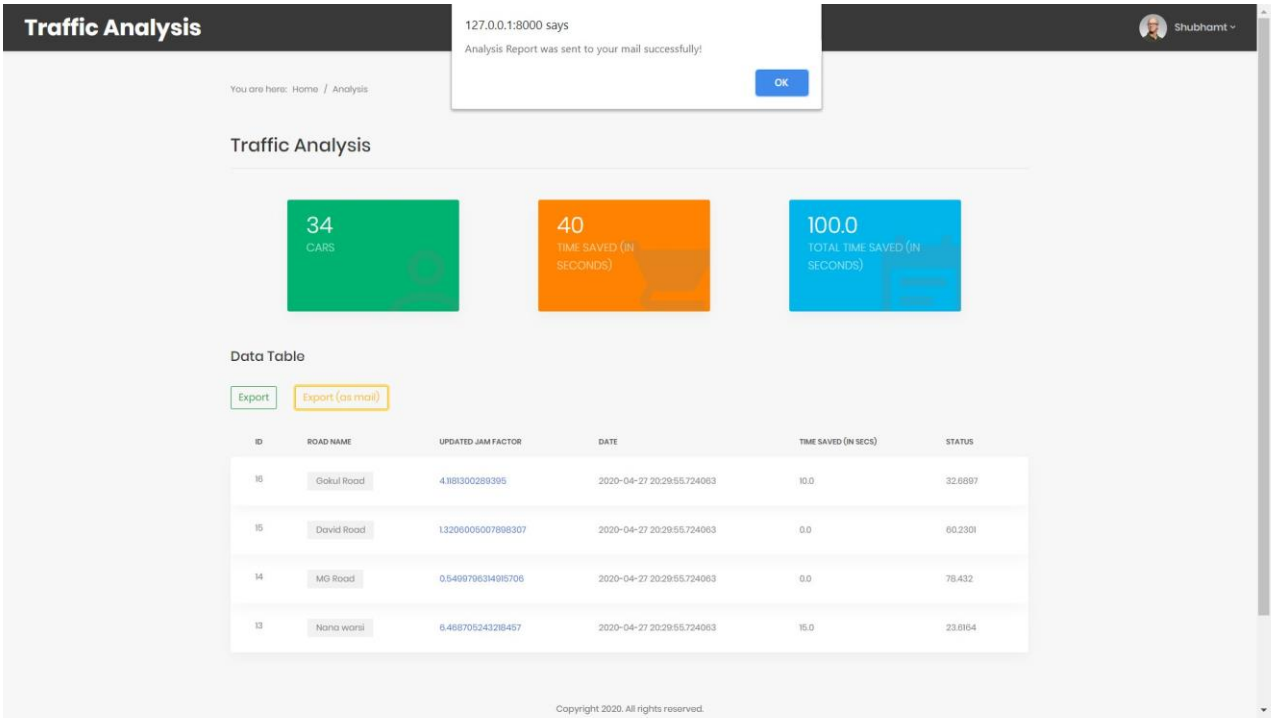


PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

	id	road_name	jam_factor	time_saved	date	status
0	1	Nana warsi	7.586776495	5	2020-04-27 20:16:34.519141	20.8621
1	2	MG Road	6.546867192	5	2020-04-27 20:16:34.519141	23.4004
2	3	David Road	3.321078924	10	2020-04-27 20:16:34.519141	37.5864
3	4	Gokul Road	2.269969896	0	2020-04-27 20:16:34.519141	46.8387
4	5	Nana warsi	6.971766743	15	2020-04-27 20:18:44.495550	22.2922
5	6	MG Road	0.286948205	0	2020-04-27 20:18:44.495550	87.4528
6	7	David Road	3.405679168	5	2020-04-27 20:18:44.495550	36.9981
7	8	Gokul Road	4.956664085	10	2020-04-27 20:18:44.495550	28.7494
8	9	Nana warsi	5.729121841	15	2020-04-27 20:20:54.494884	25.8762
9	10	MG Road	4.011682357	10	2020-04-27 20:20:54.494884	33.2686
10	11	David Road	0.399133052	0	2020-04-27 20:20:54.494884	83.3634
11	12	Gokul Road	0.882777281	0	2020-04-27 20:20:54.494884	20.9847
12	13	Nana warsi	6.468705243	15	2020-04-27 20:29:55.724063	23.6164
13	14	MG Road	0.549979631	0	2020-04-27 20:29:55.724063	78.432
14	15	David Road	1.320600501	0	2020-04-27 20:29:55.724063	60.2301
15	16	Gokul Road	4.118130029	10	2020-04-27 20:29:55.724063	32.6897

Figure 6.13: Excel sheet saved

The export to the mail gives you the option to send the excel sheet directly to the mail without the hassle of having to save it to the system.


Traffic Analysis
Shubhamt
You are here: Home / Analysis
OK
34
CARS
40
TIME SAVED (IN SECONDS)
100.0
TOTAL TIME SAVED (IN SECONDS)
Export
Export (as mail)
Copyright 2020. All rights reserved.

127.0.0.1:8000 says

Analysis Report was sent to your mail successfully!

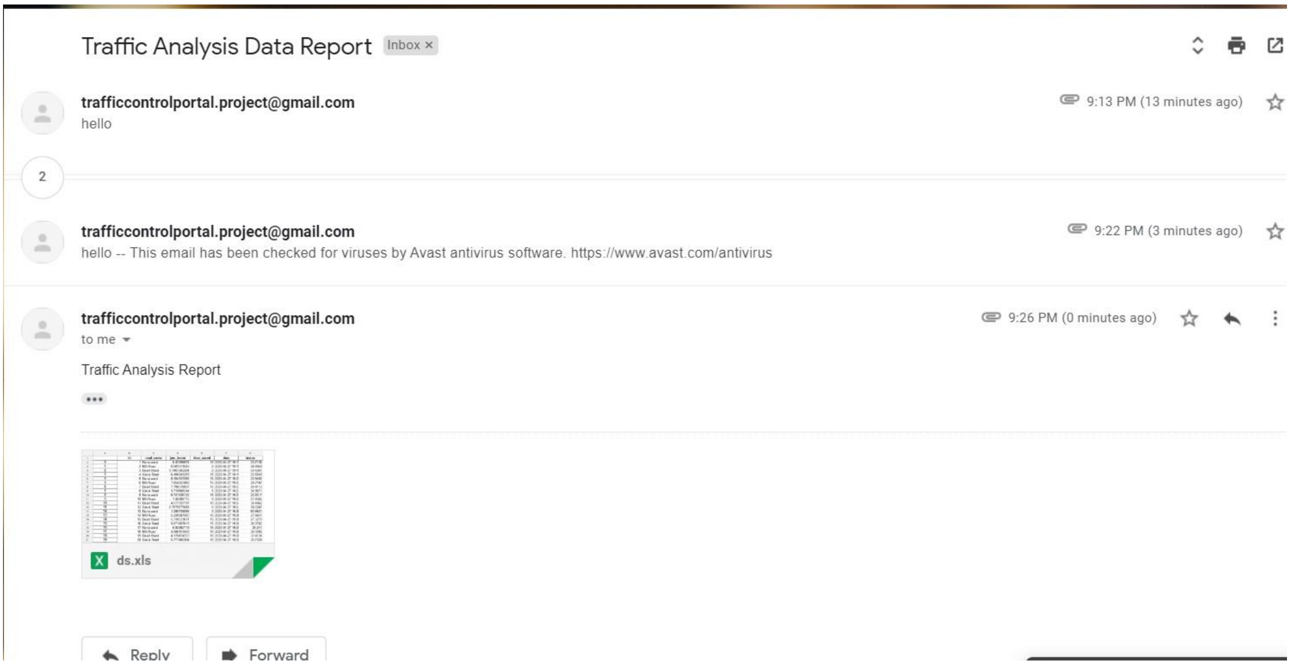
Traffic Analysis

Data Table

ID	ROAD NAME	UPDATED JAM FACTOR	DATE	TIME SAVED (IN SECS)	STATUS
16	Gokul Road	4.1181300289395	2020-04-27 20:29:55.724063	10.0	32.6897
15	David Road	1.3206005007898307	2020-04-27 20:29:55.724063	0.0	60.2301
14	MG Road	0.5499796314915706	2020-04-27 20:29:55.724063	0.0	78.432
13	Nana warsi	6.468705243218457	2020-04-27 20:29:55.724063	15.0	23.6164

Figure 6.14: Mail sent to the user

As soon as the mail is sent, an alert is issued that the mail is sent to the user.



Traffic Analysis Data Report

trafficcontrolportal.project@gmail.com 9:13 PM (13 minutes ago)
hello

trafficcontrolportal.project@gmail.com 9:22 PM (3 minutes ago)
hello -- This email has been checked for viruses by Avast antivirus software. https://www.avast.com/antivirus

trafficcontrolportal.project@gmail.com 9:26 PM (0 minutes ago)
to me

Traffic Analysis Report

Figure 6.15: Mail received by user

Chapter 7

Software Testing Document

This chapter discusses about the various test strategies that will be used. Also various test cases and features to be tested and features not be tested are discussed.

7.1 Introduction

7.1.1 System overview

Traffic blockage happens when a volume of traffic or modular split produces interest for space more prominent than the accessible road limit; this point is usually named immersion. There are various explicit conditions which cause or disturb clog; the greater part of them diminish the limit of a street at a given point or over a specific length, or increment the quantity of vehicles required for a given volume of individuals or products.

Our traffic signal management system will make use of a machine learning algorithm in order to distribute traffic evenly throughout the junction. Here, the user, after logging into the portal, will enter the road junction name based on which, the system will provide the co ordinates of that particular junction. The data returned will be real time. The system will distribute the traffic accordingly taking jam factor into consideration. Thus, our system overcomes the issues posed by image processing systems and sensor-based systems.

7.2 Test Approach

We will perform white box and black box testing on our system. In both the testing approaches, we will validate using unit testing and integration testing.

Unit Testing: In this testing all the different units will be tested and checked whether each component or module is working properly as per requirements or not. Every individual unit of program must run error free.

Integration Testing: In this approach, we will check whether all the units together as a whole work without errors and also check the flow. All the units when integrated should work without any bug.

Other than the above testing approach we will also be testing the performance and usability of the system. We will be using Performance testing and Functional testing.

Functional Testing : It will check whether the functional requirements are working properly and the algorithm provides the desired outputs.

Performance Testing: It will check the response time of the system.

7.3 Test Plan

7.3.1 Features to be tested

7.3.1.1 Data collection through HERE Maps API

These tests will check whether the data obtained through API is correct and are received without any delay.

7.3.1.2 Correct prediction of traffic

These tests will check after getting data system must predict the traffic density in the region correctly.

7.3.1.3 Provide accurate output and traffic signals

These tests will check whether the system is providing correct solution to the predicted traffic density in the region

7.3.1.4 Provide proper coordinates for given place

These tests will check that system must provide coordinates to the user about given place.

7.3.2 Features not to be tested

7.3.2.1 Portal Security

Finding vulnerabilities in the software and having strict and proper measures to ensure security and integrity of data has not been considered during the creation of Adaptive traffic control system. Since the project has been created with the submission deadline in mind and an unclear vision about its future, there hasn't been much attention paid on that area offocus.

7.3.3 Testing Tools and Environments

As we are going to follow incremental process methodology for development so testing will be done after each release, it will represent black box testing of each module, So it will go along till the completion of the project. Test tools required for testing are: Junit And Jmeter.

7.4 Test Cases

7.4.1 TC-0001

7.4.1.1 Purpose

Verify that data from API is correctly received

7.4.1.2 Inputs

JSON Data From Here Maps API

7.4.1.3 Expected Output and Pass Fail Criteria

Data should be received correctly and should be forwarded for prediction

7.4.1.4 Test Procedure

Verify that system is connected to stable internet connections and then check whether the software is successfully hitting the HERE maps API[4] to extract data correctly without any delay.

7.4.2 TC-0002

7.4.2.1 Purpose

To check that system successfully process the data and predict the density of traffic in that region

7.4.2.2 Inputs

Processed data after receiving from HERE maps API[4]

7.4.2.3 Expected Output and Pass Fail Criteria

If all the inputs are correct and system is running successfully then we will get accurate prediction.

7.4.2.4 Test Procedure

Processed data is fed to the system and then AI scripts runs on that and gives the prediction of traffic density

7.4.3 TC-0003

7.4.3.1 Purpose

To check after prediction the system must give desired output to traffic signals to reduce traffic in the area.

7.4.3.2 Inputs

Predicted data from the system

7.4.3.3 Expected Output and Pass Fail Criteria

System must give accurate and optimum results than the present one to traffic signals to control the traffic in the region

7.4.3.4 Test Procedure

Predicted data is forwarded to the next module where output is given to traffic signals and amount of traffic should reduce in the region

7.4.4 TC-0004

7.4.4.1 Purpose

To check whether the user is able to sign up and details are updated in the database as well

7.4.4.2 Inputs

Username, Email id and password

7.4.4.3 Expected Output and Pass Fail Criteria

If all the values and required fields are filled properly, then the data should be stored in database and the user should be redirected to login page

7.4.4.4 Test Procedure

Verify if user has entered legit values by comparing those values stored in a variable to some defined function as appropriate.

7.4.5 TC-0005

7.4.5.1 Purpose

To check whether the system gives notification to user after log in and attempt to login

7.4.5.2 Inputs

Login credentials- Email id and password

7.4.5.3 Expected Output and Pass Fail Criteria

User must get notification about login activity through mail

7.4.5.4 Test Procedure

A bogus user will try to login to the account using fake password.

7.4.6 TC-0006

7.4.6.1 Purpose

To check whether the system doesn't redirect the user to the main page through URL without login

7.4.6.2 Inputs

URL (main page)

7.4.6.3 Expected Output and Pass Fail Criteria

User must be redirected to the login page if he/she tries to access main page through URL

7.4.6.4 Test Procedure

Input the URL of main page and check whether it is redirecting to the login page or not

7.4.7 TC-0007

7.4.7.1 Purpose

To check if the system is returning proper coordinates for the input place

7.4.7.2 Inputs

Place name

7.4.7.3 Expected Output and Pass Fail Criteria

The coordinates should be proper when plotted on map

7.4.7.4 Test Procedure

When system returns the coordinates value, we will check the coordinates' efficiency by plotting them on Google maps.

7.4.8 TC-0008

7.4.8.1 Purpose

To check if data is fetched properly through given coordinates

7.4.8.2 Inputs

Coordinates of the place

7.4.8.3 Expected Output and Pass Fail Criteria

Traffic data of that locality should be received. The data should be in JSON format

7.4.8.4 Test Procedure

The JSON data will be verified by the code which we used to fetch the training data. The data should be similar

7.4.9 TC-0009

7.4.9.1 Purpose

To check if the Flask API created by us is working properly and fetching data without reloading the page

7.4.9.2 Inputs

Predicted data, Flask API created by us

7.4.9.3 Expected Output and Pass Fail Criteria

As soon as one cycle ends,the Flask API should fetch the predicted data without the page getting reloaded

7.4.9.4 Test Procedure

Check for page reload after one cycle ends.If it is not reloaded, then API is working properly.

7.4.10 TC-0010

7.4.10.1 Purpose

To check if the other cycle starts automatically as soon as the prior cycle of traffic timing ends.

7.4.10.2 Inputs

Predicted data, Flask API created by us

7.4.10.3 Expected Output and Pass Fail Criteria

As soon as one cycle ends, the system should fetch the data automatically without any action on page and the next cycle should begin.

7.4.10.4 Test Procedure

Start one cycle for demo basis and check if the next cycle is being called without any action of the user.

7.4.11 TC-0011

7.4.11.1 Purpose

To check if the statistics and efficiency data is getting exported as Excel sheet.

7.4.11.2 Inputs

Statistics and efficiency calculated data

7.4.11.3 Expected Output and Pass Fail Criteria

When we click on export, the data should be saved and exported in an excel sheet for the user

7.4.11.4 Test Procedure

Verify whether export onclick, exports the data to the user by clicking on it and testing.

7.4.12 TC-0012

7.4.12.1 Purpose

To check if user can logout without any hindrance

7.4.12.2 Inputs

Onclick logout action

7.4.12.3 Expected Output and Pass Fail Criteria

The user should be able to log out as soon as he clicks on it.

7.4.12.4 Test Procedure

Onclick logout action will be performed and the case will be tested.

7.4.13 TC-0013

7.4.13.1 Purpose

To check if the stop button on analysis will lead to stopping of simulation.

7.4.13.2 Inputs

Onclick stop action

7.4.13.3 Expected Output and Pass Fail Criteria

The simulation should stop and no more cycles should take place

7.4.13.4 Test Procedure

Onclick stop action will be performed and the case will be tested.

7.5 Test Logs

Table 7.1: Test logs

Sr.no	Test Case ID	Description	Input	Expected Output	Actual Output	Criteria
1	TC-0001	Verify that data from api is correctly received	JSON data from Here maps API	Received data is correct	Received data is correct	PASS
2	TC-0002	Verify that data is successfully stored in database	Data in JSON format from Here maps API	Data is successfully inserted into database	Data is successfully inserted into database	PASS
3	TC-0003	To check after prediction the system must give desired output to traffic signals to reduce traffic in the area.	Predicted data from the system	System must give accurate and optimum results than the present one to traffic signals to control the traffic in the region	Returns optimal solution for the given jam factors	PASS
4	TC-0004	To check whether the user is able to sign up and details are updated in the databases as well	Username, email id and password	If all the values and required fields are filled properly, then the data should be stored in database and the user should be redirected to login page	The sign up details are reflected on the database only if legit values are entered and the user is redirected to login page	PASS
5	TC-0005	To check whether the system gives notification to user after log in and attempt to login	Login credentials-email id and password	User must get notification about login activity through mail	User get notification about login activity through mail	PASS

Continued on next page

Table 7.1 – continued from previous page

Sr.no	Test Case ID	Description	Input	Expected Output	Actual Output	Criteria
6	TC-0006	To check whether the system doesn't redirect the user to the main page through URL without login	URL (main page)	User must be redirected to the login page if he/she tries to access main page through URL without login	User can't access the portal through URL if not logged in	PASS
7	TC-0007	To check if the system is returning proper coordinates for the input place	Place name	The coordinates should be proper when plotted on map	Gives the same location for the given coordinates on Google and HERE maps	PASS
8	TC-0008	To check if data is fetched properly through given coordinates	Coordinates of the place	The JSON data will be verified by the code which we used to fetch the training data. The data should be similar	The JSON data fetched is verified and is same as on the website as well	PASS
9	TC-0009	To check if the Flask API created by us is working properly and fetching data without reloading the page	Predicted data, Flask API created by us	As soon as one cycle ends,the Flask API should fetch the predicted data without the page getting reloaded	The page doesn't reload and the data is fetched properly using Flask API	PASS
10	TC-0010	To check if the other cycle starts automatically as soon as the prior cycle of traffic timing ends.	Predicted data, Flask API created by us	As soon as one cycle ends, the system should fetch the data automatically without any action on page and the next cycle should begin.	The next cycle begins as soon as the prior one ends and no onpage action is done	PASS

Continued on next page

Table 7.1 – continued from previous page

Sr.no	Test Case ID	Description	Input	Expected Output	Actual Output	Criteria
11	TC-0011	To check if the statistics and efficiency data is getting exported as Excel sheet.	Statistics and efficiency calculated data	When we click on export, the data should be saved and exported in an excel sheet for the user	The excel sheet is exported with the data	PASS
12	TC-0012	Verify if the user can log out without any hindrance	Onclick logout action	On clicking the log out option, the user should be able to log out without any issues	On clicking the log out option, the user logs out without any issues	PASS
13	TC-0013	To check if the stop button on analysis will lead to stopping of simulation.	Onclick stop action	the simulation should stop and no more cycles should take place	On clicking the Stop button, the simulation stops without any issues	PASS

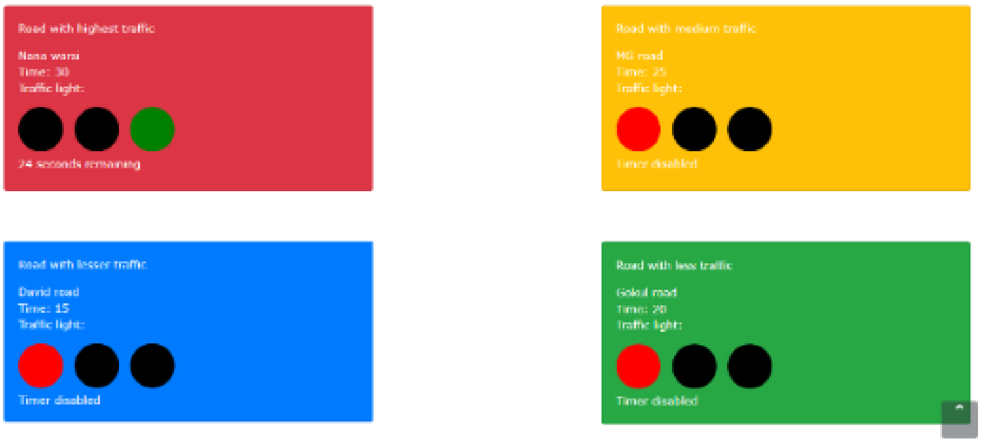
Chapter 8

Results and discussion

This chapter discusses about the various test strategies that will be used. Also it gives the output of various algorithms we used to reach the conclusion. And which one of the algorithm we used finally.

To execute our objectives and targets we first concentrate information from HERE Maps API, which gives us the traffic density information as JSON or XML position. We get different highlights with respect to the traffic out and about, for example, the normal speed of any vehicle on that specific street, the length of the street, the most extreme speed of any vehicle out and about and the jam factor (the traffic density) on that stretch of street.

The following are the screenshots of the working of the system:



Road with highest traffic

Nana warsi
Time: 30
Traffic light:

⚫ ⚫ 🟢

24 seconds remaining

Road with medium traffic

MG road
Time: 25
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with lesser traffic

David road
Time: 15
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Road with less traffic

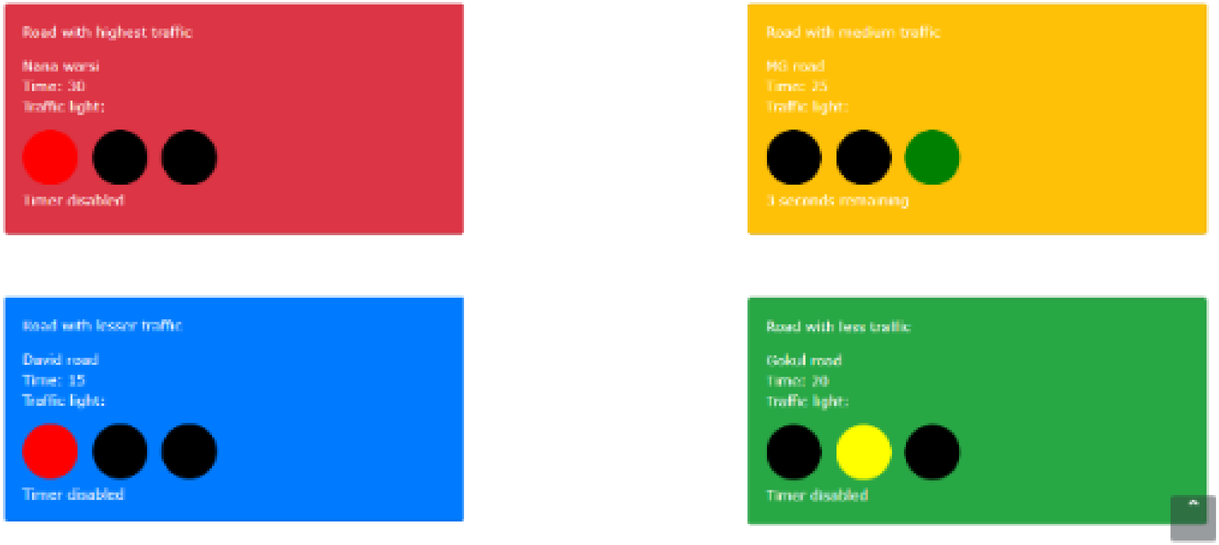
Gokul road
Time: 20
Traffic light:

🔴 ⚫ ⚫

Timer disabled

Figure 8.1: Simulation started

When there are 3 seconds left, the next road’s signal gets yellow.



Road with highest traffic

Nana warsi
Time: 30
Traffic light:

Timer disabled

Road with medium traffic

MG road
Time: 25
Traffic light:

3 seconds remaining

Road with lesser traffic

David road
Time: 15
Traffic light:

Timer disabled

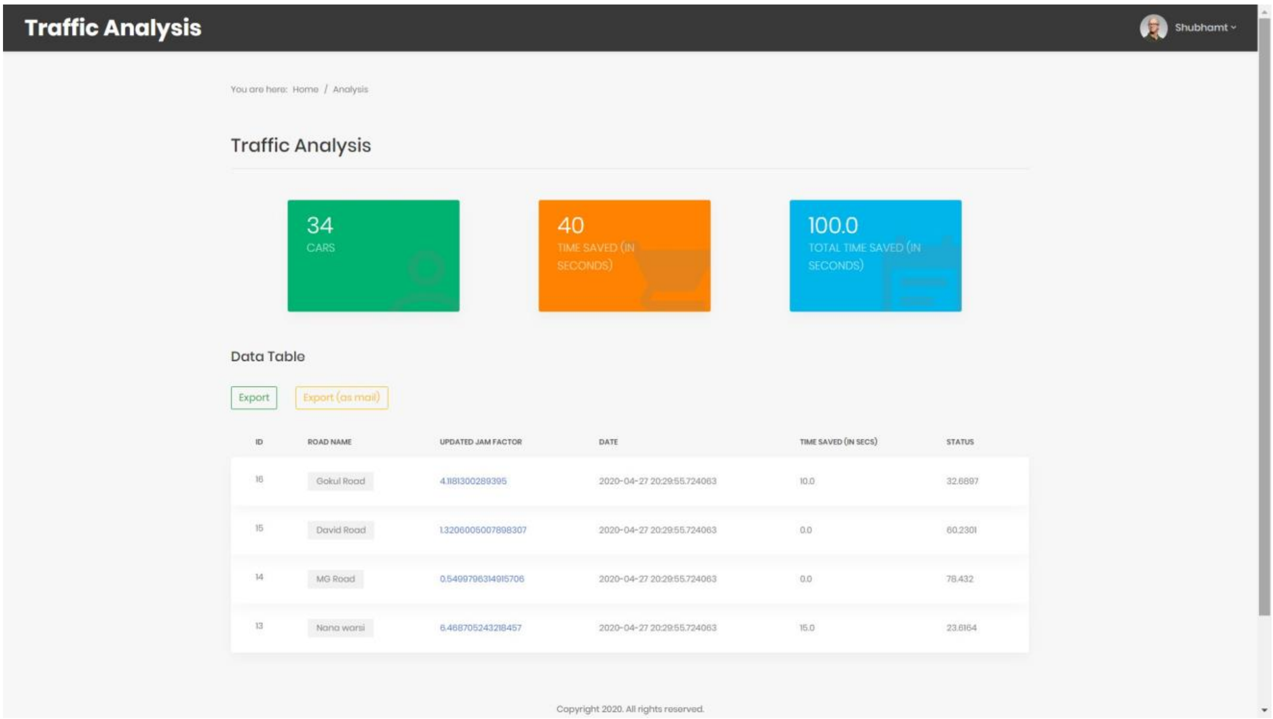
Road with less traffic

Gokul road
Time: 20
Traffic light:

Timer disabled

Figure 8.2: Simulation midway

The statistics from our system are also calculated.The cars box shows us the total number of cars that left the road and didn't stuck in traffic because of our system. The time saved depicts that how much the time was saved in last cycle whereas the total time saved shows time saved in total by our system. Every road has some optimized jam factor afterwards, which is shown in the table. Also, the percentage of efficiency is calculated for the same.
The boxes at the top show the total statistics of our system.



You are here: Home / Analysis

Traffic Analysis

34
CARS

40
TIME SAVED (IN SECONDS)

100.0
TOTAL TIME SAVED (IN SECONDS)

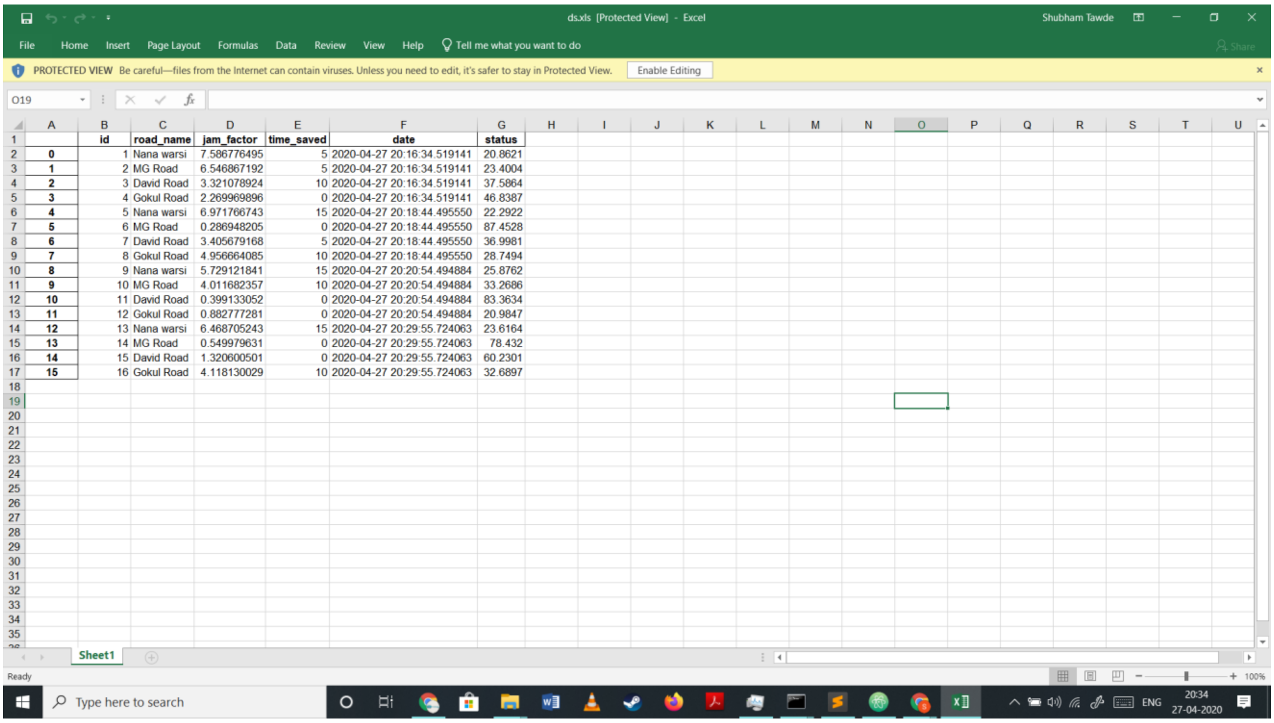
Data Table

ID	ROAD NAME	UPDATED JAM FACTOR	DATE	TIME SAVED (IN SECS)	STATUS
16	Gokul Road	4.1181300289395	2020-04-27 20:29:55.724063	10.0	32.6897
15	David Road	1.3206005007898307	2020-04-27 20:29:55.724063	0.0	60.2301
14	MG Road	0.5499796314915706	2020-04-27 20:29:55.724063	0.0	78.432
13	Nana warsi	6.468705243218457	2020-04-27 20:29:55.724063	15.0	23.6164

Copyright 2020. All rights reserved.

Figure 8.3: Statistics and Efficiency factor page

The export button gives you the option to save the data to an excel sheet. The excel sheet contains the road name, jam factor, time saved, timestamp, efficiency percentage. The export to mail just mails you the excel sheet to avoid the hassle of having to save it on the disk.



ds.xls [Protected View] - Excel
Shubham Tawde
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
Share
PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.
Enable Editing
O19
Sheet1
Ready
100%
Type here to search
ENG
20:34
27-04-2020

	id	road_name	jam_factor	time_saved	date	status
0	1	Nana warsi	7.586776495	5	2020-04-27 20:16:34.519141	20.8621
1	2	MG Road	6.546867192	5	2020-04-27 20:16:34.519141	23.4004
2	3	David Road	3.321078924	10	2020-04-27 20:16:34.519141	37.5864
3	4	Gokul Road	2.269969896	0	2020-04-27 20:16:34.519141	46.8387
4	5	Nana warsi	6.971766743	15	2020-04-27 20:18:44.495550	22.2922
5	6	MG Road	0.286948205	0	2020-04-27 20:18:44.495550	87.4528
6	7	David Road	3.405679168	5	2020-04-27 20:18:44.495550	36.9981
7	8	Gokul Road	4.956664085	10	2020-04-27 20:18:44.495550	28.7494
8	9	Nana warsi	5.729121841	15	2020-04-27 20:20:54.494884	25.8762
9	10	MG Road	4.011682357	10	2020-04-27 20:20:54.494884	33.2686
10	11	David Road	0.399133052	0	2020-04-27 20:20:54.494884	83.3634
11	12	Gokul Road	0.882777281	0	2020-04-27 20:20:54.494884	20.9847
12	13	Nana warsi	6.468705243	15	2020-04-27 20:29:55.724063	23.6164
13	14	MG Road	0.549979631	0	2020-04-27 20:29:55.724063	78.432
14	15	David Road	1.320600501	0	2020-04-27 20:29:55.724063	60.2301
15	16	Gokul Road	4.118130029	10	2020-04-27 20:29:55.724063	32.6897

Figure 8.4: Excel sheet saved

Chapter 9

Conclusion and Future Work

This chapter mainly compares and contrast the past and present scenarios of traffic signal management. Also scope of the project and how the project can expanded is discussed in this chapter.

9.1 Conclusion

Thus, our fundamental aim is to control the traffic flags powerfully dependent on the present traffic density and considering the memorable traffic density esteems. Among the streets that we have considered, the street having the most noteworthy jam factor (traffic density) will be viewed as first and in our recreation run the jam factor will be decreased dependent on the surmised normal speed of any vehicle on that specific street and the traffic on the ensuing street. Time for a specific activity, ie, a red light or a green light will be given dependent on different elements, basically the jam factor. Accordingly, with the expanding traffic in our everyday life, the need to oversee it has become a need now. Our model gives an ideal answer for the intersections at top occasions just as different occasions. In this venture, we have proposed a strategy to forestall blockage of traffic by wiping out the customary picture handling technique utilized by the past frameworks and by controlling the traffic lights powerfully.

9.2 Future Work

As far as the future work is concerned, our system would also work towards including various other pivotal factors such as climate, accidents, festivals and curfews to determine traffic clog and work on it simultaneously. When the system gets implemented successfully, with the inclusion of all the required wholesome factors mentioned above, the government can make use of this system without needing to bother with complex processing and equipments that would otherwise be required in image processing systems and sensors in order to control the traffic congestion. Apart from this, the government need not spend even a single penny, as this work is based entirely on algorithms and removes the necessity of image processing and sensors which would otherwise mint out unnecessary budget.