**IBM NALAIYA THIRAN PROJECT REPORT TEAM ID - PNT2022TMID02365 IBM NALAIYA THIRAN 2022-23 PROJECT REPORT**

**SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY**

**TEAM ID - PNT2022TMID02365**

**Project Report Format**

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Node Red
   2. MIT App
8. **TESTING**
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

**1. INTRODUCTION**

**1.1 Project Overview**

The main goal of the project is to develop a smart connected sign board that provides speed limits based on the current status of the temperature, weather, and climate from OpenWeatherMap, as well as to indicate the school zone and hospital zone, which determines the use of Horn.

**1.2 Purpose**

* The main purpose of this project that is to keep people safe, and help to communicate messages to drivers as well as pedestrians that can maintain order and reduce accidents.
* To replace the static sign board, smart connected sign boards are used.
* Speed can be adjusted based on the weather condition.
* School zone and hospital zone are indicated in order to reduce the accidents.

1. **LITERATURE SURVEY**

**2.1 Existing Problem**

* Static sign board cannot update automaticaly based on weather condition which leads to accidents.
* School zone are shown for the drivers should watch out for children, reduce speed, and obey any signals from a crossing guard.
* Damage in static sign board such as pain deterioration, folded sheet and pole bent can lead to many accidents.

**2.2 References**

1. Mrs. Devershi Pallavi Bhatt and Mr. Manish Tiwari, in their paper titled, "Smart  Traffic Sign Boards (STSB) for Smart Cities", 2nd smart cities symposium, 24-26  March, 2019, University of Bahrain, explained how Smart sign boards with various  features can be built for modern smart cities. According to them, Smart Traffic  Management is a system to monitor and control traffic signals using sensors to regulate  traffic flow and avoid congestion for smooth traffic flow. The authors proposed a  framework of Smart Traffic Sign Boards (STSB) in this paper, which can communicate  with the system deployed in all vehicles to make drivers aware of various road safety  signs.

1. Mr. Abhishek Rai and Mr. Farooque Azam, in their paper titled, “Smart Speed  Limit Sign Board for Changing Weather Conditions”, International Journal of  Computer Sciences and Engineering, Vol.-7, 14th May 2019, explained how Smart  speed limit sign boards can be built using Iot technologies. Their project's main goal is  to contribute to smart transportation based on weather data, which can help reduce  congestion and accidents by making speed limit sign boards adaptive. The board which  they described, is not limited to displaying speed limits and can be used for other signs  as well.

1. Mr. Ching-Hao Lai and Mr. Chia-Chen Yu, in their paper titled, “An Efficient Real Time Traffic Sign Recognition System for Intelligent Vehicles with Smart  Phones”, 2010, International Conference on Technologies and Applications of  Artificial Intelligence. This project's main goal is to contribute to smart transportation  based on weather data, which can help reduce congestion and accidents by making  speed limit sign boards adaptive. This board is not limited to displaying speed limits  and can be used for other signs as well.

1. Mr. Abd-Elhamid M. Taha, in his paper titled, “An IoT Architecture for Assessing  Road Safety in Smart Cities”, Published on 19 November 2018, Wireless  Communications and Mobile Computing Volume 2018, Article ID 8214989. The Architecture proposed by him, involves a novel use of machine learning as part of its  road safety assessment core. This application facilitates assessments that are both  dynamic and robust. He also showcased an application of the developed core aimed at  safety-based route planning in smart cities.
2. Mr. Ricardo Jorge Fernandes, in his paper titled, “VANET-Enabled In-Vehicle  Traffic Signs”, Published on June 2009, University of porto, provides us a broad  overview of Identifying challenges and studying the feasibility of VANET-enabled in vehicle traffic signs in real environments. In this Paper, the author extended a state-of the-art microscopic VANET simulator to emulate virtual traffic signs, and thus,  evaluate their behaviour and feasibility.

1. Mr. Abdul Kadar Muhammad Masum, Mr. Kalim Amzad Chy, Mr. Iaamanur  Rahman, in his paper titled, “An Internet of Things (IoT) based Smart Traffic Management System: A Context of Bangladesh”, Published on 27-28 October  2018, 2nd Int. Conf. on Innovations in Science, proposed a real-time traffic  management system (TMS) using the Internet of Things (IoT) and data analytics. After  analysing the ultrasonic sensor data, system controller sets traffic signal time by traffic  management algorithm and also sends data to a cloud server through a Wi-Fi module.  The proposed system can predict probable traffic congestion in the intersection point.  If an emergency vehicle is detected, it gives priority, i.e. high signal duration to pass  the intersection. In case of the signal violation, the system can identify the vehicle and  charge a fine that is paid through Traffic Wallet mobile app.
2. Mr. Seung Byum Seo, Mr. Pamul Yadav, Mr. Dhananjay Singh, in their paper  titled, “LoRa based architecture for smart town traffic management system”,  Published on 07 November 2020, Springer Science+Business Media, LLC, part of  Springer Nature 2020, proposed various mechanisms for each step of the process, for  vehicle detection they use inductive loop sensor. Also, they have proposed a deep  learning-based image processing centered vehicle detection system. Image processing  techniques provide an efficient mechanism to recognize the vehicles on the road in real time. Neural Network algorithms are trained to classify vehicles such as cars, trucks,  buses and record the number of vehicles with improved accuracy.

1. Mr. Dries Naudts, Mr. Vasilis Maglogiannis, Mr. Daniel van den Akker, in their  paper titled, “Vehicular Communication Management Framework: A Flexible  Hybrid Connectivity Platform for CCAM Services”, Published on 22 March 2021,  Future Internet 2021. In this paper, they proposed a novel vehicular communication  management framework (CAMINO), which incorporates flexible support for both  short-range direct and long-range cellular technologies and offers built-in Cooperative  Intelligent Transport Systems’ (C-ITS) services for experimental validation in real-life  settings. Moreover, integration with vehicle and infrastructure sensors/actuators and  external services is enabled using a Distributed Uniform Streaming (DUST)  framework.

1. Mr. Da Zhang, Mr. Mansur R. Kabuka, In the paper titled, “Combining weather  condition data to predict traffic flow: a GRU-based deep learning approach”, Published on 1st March 2018, Department of Electrical and Computer  Engineering, University of Miami, Coral Gables, FL, USA, said, this study applies  gated recurrent neural network to predict urban traffic flow considering weather  conditions. Running results show that, under the review of weather influences, their  method improves predictive accuracy and also decreases the prediction error rate. To  their best knowledge, this is the first time that traffic flow is predicted in urban freeways  in this particular way. This study examines it with respect to extensive weather  influence under gated recurrent unit-based deep learning framework.

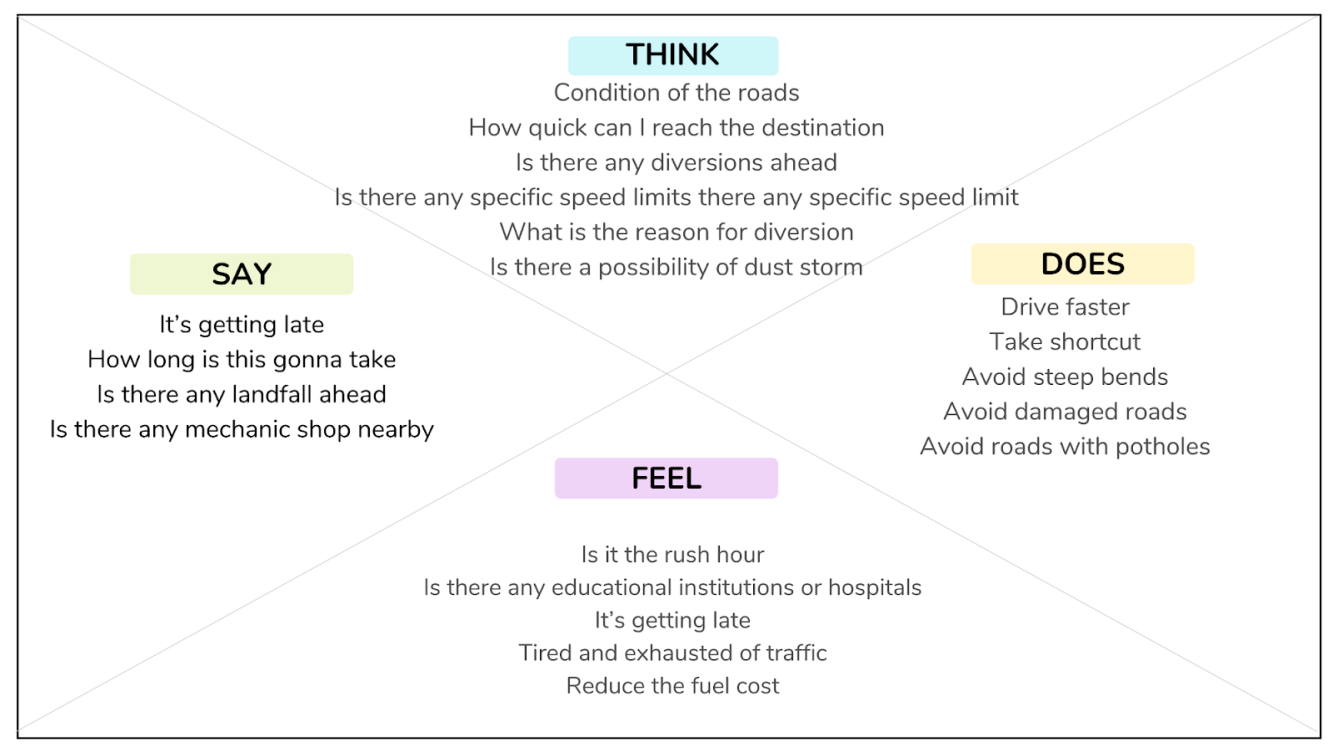
1. Mr. Muhammed O. Sayin, Mr. Chung-Wei Lin, Mr. Eunsuk Kang, in their paper  titled, “Eunsuk Kang”, Published on 3rd June 2019, Life Fellow, IEEE, they  proposed a game theoretical adversarial intervention detection mechanism for reliable  smart road signs. A future trend in intelligent transportation systems is “smart road signs” that incorporate smart codes on their surface to provide more detailed  information to smart vehicles. Such smart codes make road sign classification problem  aligned with communication settings more than conventional classification. This  enables us to integrate well-established results in communication theory, e.g., error correction methods, into road sign classification problem.

**2.3 Problem Statement Definition**

To replace static sign boards ,smart connected sign boards are used . These smart sign boards get speed limitations from weather API and update the changes automatically in the web app. Based on weather conditions the speed may increase or decrease and based on traffic and fatal situations , the road signs are displayed.

1. **IDEATION AND PROPOSED SOLUTION**

**3.1 Empathy map canvas**



**3.2 Ideation & Brainstorming**

**IDEA 1**

Internet Connectivity for the Smart sign board for road safety application employs an esp32 board that enables remote control and easy real time access to the system. There are several parameters that can be considered for a road safety system among which weather plays a major role. Real-time weather characteristics such as humidity, precipitation, temperature etc.,can be monitored by using the industrial standard sensors. These sensors are interconnected as a sensor network. A city can be divided into many zones and the sensors will be placed in different locations. Data from all the zones is sent to the cloud and appropriate preprocessing of the data will be done followed by further analysis of data in order to make the weather prediction.Based on the data acquired and the analyzed result the speed limit for the different areas will be automatically updated. The traffic signals employ a camera that continuously monitors the traffic and intelligently changes the signal depending upon the intensity of the traffic on each side. In addition, a sensor is attached to all the vehicles to monitor the condition of the road. The road condition is updated to the cloud and depending on the condition, the diversions are introduced in the path.

**IDEA 2**

APIs let your product or service communicate with other products and services without having to know how they’re implemented.By using weather API, we can get the data about past climatic conditions, and make some ambiguous future predictions about the climatic conditions using various prediction models. According to the predicted results, we can give an alert sign in the smart sign board.So, using such APIs for gathering information will be a reliable form of previous,past and future data.

In addition to that, we can use a reliable camera for detecting traffic conditions(providing traffic signs accordingly), the camera can also be used for detecting signs like zebra crossing( for providing speed limit signs).APIs such as ArcGIS API may also be used to

obtain information about adjacent school or hospital zones, analyze the information, and display speed restriction signs accordingly. So, these are some of the ideas which can be implemented in smart sign boards.

**IDEA 3**

Google Maps API may be used to acquire access to information about traffic congestions, and information from the traffic layer can be used to gain access to real-time traffic information.This inturn assists us in dividing and diverting the traffic and also comes in handy when dealing with emergency situations.For example, if an ambulance needs to get to a hospital swiftly, traffic information may be utilized to discover faster routes, reducing time delays and allowing patients to be delivered in the shortest possible amount of time.Google APIs can also be used to locate the nearest hospitals and all the possible routes to reach the same.

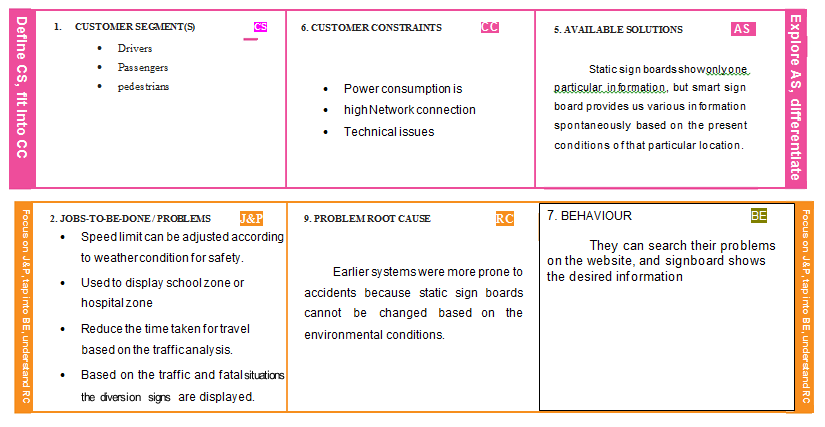
**IDEA 4**

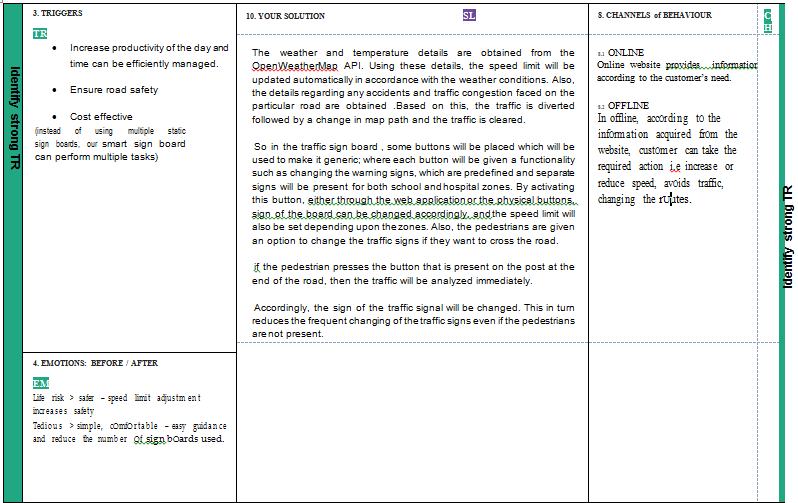
Sign boards can be used to indicate school zones and hospital zones and thereby alter the speed range using location information. From the web application, we can get the speed limitations which updates automatically to the driver. We may also acquire information about any construction on the road and during fatal situations, the diversion signs are displayed. The pedestrians are given access to control the crossing sign by pressing a button when required. When a pedestrian presses the button the traffic is sensed on each side and if less traffic is detected the red signal is given for the vehicles. In this way the workload on the camera in the traffic signal can be reduced .

**3.3 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Parameter** | **Description** |
| 1. | Problem Statement | To replace the static signboards, smart connected sign boards are used.  These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.  Based on the weather changes the speed may increase or decrease  Based on the traffic and fatal situations the diversion signs are displayed.  Guide(Schools), Warning and Service(Hospitals, Restaurant) signs are also displayed accordingly.  Different modes of operations can be selected with the help of buttons. |
| 2. | Idea Description | The weather and temperature details are obtained from the OpenWeatherMap API. Using these details, the speed limit will be updated automatically in accordance with the weather conditions. Also, the details regarding any accidents and traffic congestion faced on the particular road are obtained .Based on this,the traffic is diverted followed by a change in map path and the traffic is cleared. So in the traffic sign board , some buttons will be placed which will be used to make it generic; where each button will be given a functionality such as changing the warning signs, which are predefined and separate signs will be present for both school and hospital zones.By activating this button, either through the web application or the physical buttons, sign of the board can be changed accordingly, and the speed limit will also be set depending upon the zones. Also, the pedestrians are given an option to change the traffic signs if they want to cross the road. If the pedestrian presses the button that is present on the post at the end of the road, then the traffic will be analyzed immediately. Accordingly, the sign of the traffic signal will be changed. This inturn reduces the frequent changing of the traffic signs even if the pedestrians are not present. |
| 3. | Novelty | Generic Sign board for all applications that uses both buttons and web service for updation  Pedestrians are given the access to request the sign change of the signal to cross the road. |
| 4. | Customer  Satisfaction | Diversion reasons will be displayed  If there is no traffic, pedestrians can cross the street without waiting. Customer can reach the destination before the expected time |
| 5. | Business Model | Since APIs are used to actively monitor the customer's environment, this project employs a business strategy in which revenue will be generated on the basis of the length of time in which the customers actively interact with the product.  This product is aimed to be free of cost to the public, but the revenue will be generated by selling this product to the government at a low cost, so there will be less accidents and the public will be aware of the discrepancies or accidents in the particular road. The public will also gain all the information about the road, even if they are checking for an alternate path because of some mishaps that happen on the roads and these functionalities will increase the value of the product in the global market. |
| 6. | Scalability of the Solution | In the future, if any update is required either on the hardware or software side, it can be easily implemented. The hardware components can be directly interfaced with the microcontroller and small modifications can be made in the programming of the existing product. In case of the software, the website application has to be updated with the additional functionality by creating a new section for the updated hardware. So this will not affect the existing functionality of the product and new functionality can be easily integrated. In addition, a separate circuit will be kept along with the hardware to detect any problem which informs the web application. Also a notification will be sent to the product service department. |

**3.4 Problem Solution fit**

****



**4. REQUIREMENT ANALYSIS**

**4.1 Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Traffic Detection | The traffic data is necessary for diverting the route. The traffic data is obtained from the camera. |
| FR-2 | Dynamic Display | The display is automatically updated based on the traffic data and weather conditions to reduce the manual updating of display. |
| FR-3 | Weather Monitoring | Speed limit is set based on the weather conditions |
| FR-4 | Manual updation | Buttons are used to navigate across various processes in the smart sign board. |
| FR-5 | Website and Cloud | All the data is collected and processed on the cloud and the web application provides the user interface. |

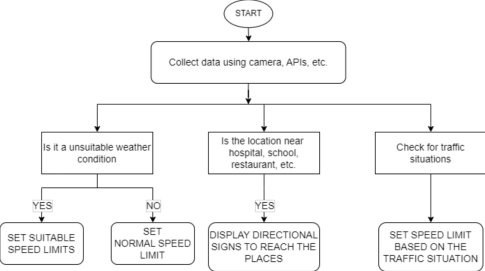
**4.2 Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The usage of the web application must be simple and user friendly. |
| NFR-2 | **Security** | The cloud access and other services are restricted to the government officials and all the updation can be done from the remote location. The username and key should be confidential. |
| NFR-3 | **Reliability** | The information sent by the sensors should be accurate. The hardware components used for the system and the code should be written in such a way that the time taken to send the data is minimum. |
| NFR-4  NFR-5 | **Performance**  **Availability** | Traffic and weather condition values should be updated frequently.  The product must be easily available to the customer. |
| NFR-6 | **Scalability** | The project must have the ability to increase or decrease in performance and cost in response to changes in application and system processing demands. |

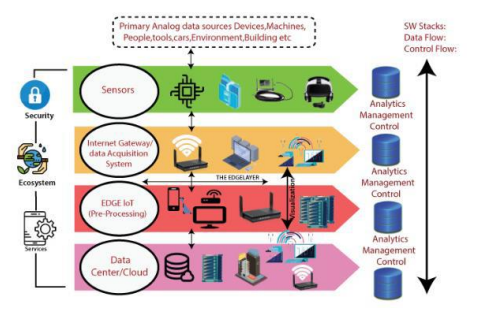
**5.PROJECT DESIGN**

**5.1 Data Flow Diagrams**

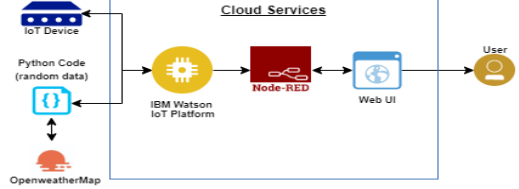


**5.2 Solution Architecture**

* Speed limit is adjusted automatically according to weather condition for safety.
* These smart sign boards are monitored remotely using web and mobile application weather API and update automatically.
* Based on the traffic and fatal situations the diversion signs are displayed.
* Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.



**5.3 Technical Architecture:**

****

**Table-1 : Components & Technologies:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Component Description** | **Technology** |
| 1. | User Interface How user interacts with application e.gWeb UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
| 2. | Application Logic-1 To define steps and procedures for various operations performed in the process | Python script |
| 3. | Application Logic-2 To add and manage iot devices, control access to **IoT** service, and monitor the usage. | IBM WATSON IoT Platform |
| 4. | Application Logic-3 To provide a connection between API , cloud, hardware,etc. | Node-RED |
| 5. | Database Data Type, Configurations etc. | MySQL, NoSQL, etc. |
| 6. | Cloud Database Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
| 7. | File Storage File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
| 8. | External API-1 OpenWeatherMap API is used to obtain weather and temperature details | HTTP |
| 9. | Infrastructure (Server / Cloud) Application Deployment on Local System / Cloud Local Server Configuration:  Cloud Server Configuration : | Cloud Foundry |

**Table-2: Application Characteristics:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristics Description** | **Technology** |
| 1. | Open-Source Frameworks Used for weather forcasting | Wecast |
| 2.  3. | Security Implementations It monitors and filters incoming and outgoing network traffic for security.  Scalable Architecture Justify the scalability of architecture (3 – tier, Micro-services) | Firewalls, etc.  IoT, internet. |
| 4. | Availability The service of the product will be available 24/7 | APIs,cloud,etc. |

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristics Description** | **Technology** |
| 5. | Performance It is a metric that measures the throughput of a system. | Request handling |

**References:**

**https://c4model.com/**

**https://www.ibm.com/cloud/architecture**

[**https://openweathermap.org/api**](https://openweathermap.org/api)

**5.4 User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement**  **(Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (Mobile User) | Registration | usn-1 | I can have access to the speed  constraints utilizing the weather data | I need to know the  speed limit and weather condition | High | Sprint-1 |
|  |  | usn-2 | The school and the hospital zone is manually updated using the buttons in the signboard manually. | Is it a school leaving time or starting time Is it a hospital zone or school zone | Medium | Sprint-2 |
|  |  | usn-3 | Based on the traffic and weather condition the speed limit is adjusted automatically | Speed data is available in the website | Low | Sprint-1 |
|  |  | usn-4 | When the traffic is intense or if any accidents or construction occurs in the road then the diversion sign is showed | I can take diversion by the information we get from the website | Medium | Sprint-1 |
|  | Login | usn-5 | The login credentials for the cloud is only with the government officials and the user can use the website to see the traffic details and diversion in the particular road | I must be able to view the traffic and diversion details from the website | Low | Sprint-2 |
| Customer  (web user) | Data generation | usn-6 | Data is obtained from the sensors and it is sent to the website | I must receive accurate sensor data for  processing | High | Sprint-1 |

**6. Project Planning and Scheduling**

**6.1 Sprint Planning & Estimation**

**Product Backlog, Sprint Schedule, and Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by  entering my email, password, and confirming  my password. | 2 | High | Kanish |
| Sprint-1 |  | USN-2 | As a user, I will receive confirmation email once  I have registered for the application | 1 | High | Jegadeeswar |
| Sprint-1 |  | USN-3 | As a user, I can register for the application  through Facebook | 2 | Low | Jayapreethi |
| Sprint-1 |  | USN-4 | As a user, I can register for the application  through Gmail | 2 | Medium | Lavanya |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by  entering email & password | 1 | High | Lavanya  Jegadeeswar |
| Sprint-1 | Dashboard | USN-6 | As a user, I can log into the application by  entering email & password and access all the  resources and services available | 2 | High | Jayapreethi  Kanish |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-2 | Login | USN-1 | As a weather data controller, I log into my  profile and start monitoring the weather updates | 3 | High | Kanish  Jegadeeswar |
| Sprint-2 | Dashboard | USN-2 | I receive all the information about weather from  web from weather API. Whenever there is  change in weather, corresponding updates are  made on sign boards. | 2 | Medium | Jegadeeswar Kanish  Lavanya  Jayapreethi |
| Sprint-3 | Login | USN-1 | As a image controller, I keep note of all the  images received from various areas and detect  traffic in that particular area. | 3 | High | Lavanya  Jegadeeswar Jayapreethi  Kanish |
| Sprint-3 | Dashboard | USN-2 | With the traffic, updates I change the status of  sign board as “take diversion”. | 2 | Medium | Lavanya  Jayapreeti |
| Sprint-4 | Login | USN-1 | As a zonal officer, I ensure that boards near  school display “slow down” and near hospitals  display “no horn”. | 3 | High | Lavanya  Kanish |
| Sprint-4 | Login | USN-1 | As an administrator, I ensure that all  departments work co-ordinated and ensure the  accuracy and efficiency. | 2 | Medium | Jegadeeswar Jayapreethi |

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story**  **Points** | **Duration** | **Sprint Start Date** | **Sprint End Date**  **(Planned)** | **Story Points**  **Completed (as on  Planned End Date)** | **Sprint Release Date**  **(Actual)** |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

**Velocity:**

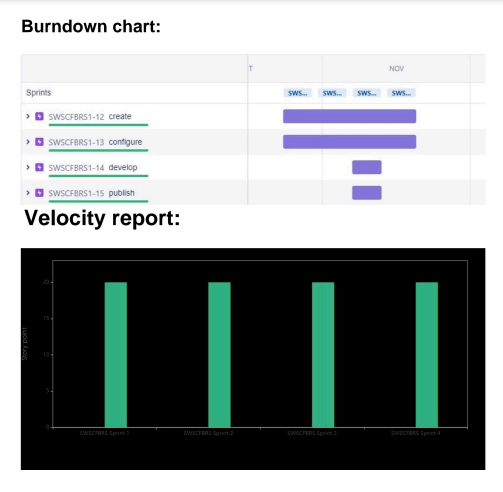
Imagine we have a 10 day sprint duration, and the velocity of the team is 20  
(points per sprint). Let’s calculate the team’s average velocity per iteration unit

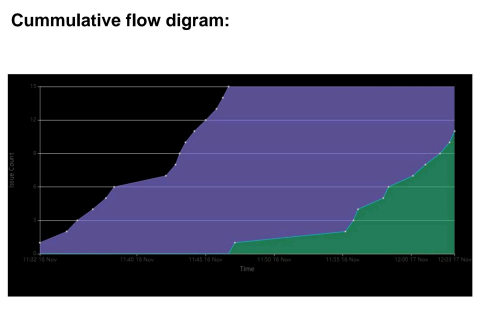
**6.2 Sprint Delivery**

**Milestone and Sprint Delivery**

|  |  |  |
| --- | --- | --- |
| TITLE | DESCRIPTION | DATE |
| Literature Survey&  Information Gathering | A literature review is a  comprehensive summary  of previous researches on  the topic. The literature  review surveys scholarly  articles, books, and other  sources relevant to a  particular area of  research. | 1 September 2022 |
| Prepare Empathy Map | An empathy map is a  collaborative tool teams  can use to gain a deeper  insight into their  customers. It helps us to  understand the customers’  pain, gain and difficulties  from their point of view. | 7 September 2022 |
| Ideation  Brainstorming | Brainstorming is a  group problem-solving  method that helped us to  gather and organize  various ideas and  thoughts from team  members. | 16 September 2022 |
| Define Problem  statement | The Customer Problem  Statement helps us to  focus on what matters to  create experiences people  will love. A well-articulated  customer problem  statement allowed us to  find the ideal solution for  the challenges customers  face. | 19 September 2022 |
| Problem Solution Fit | It helped us understand  and analyse all the  thoughts of our customer,  their choice of options,  problems, root cause,  behaviour and emotions. | 14 October 2022 |
| Proposed solution | It helped us analyse and  examine our solution more  in the grounds of  uniqueness, social impact,  business model, scalability  etc. | 19 October 2022 |
| Solution Architecture | Solution architecture is a  complex process – with  many sub-processes – that bridges the gap  between business  problems and technology  solutions. It helped us  understand the features  and components used to  complete the project. | 27 October 2022 |
| Customer journey map | It helped to analyse the  various steps, interactions,  goals and motivation,  positives, negatives and  opportunities. | 15 October 2022 |
| Solution requirements | It briefs about functional  and non-functional  requirements. It involves  the various steps in the  entire process. It also  specifies features  usability, security,  reliability, performance,  availability and scalability. | 12 October 2022 |
| Technology stack | A tech stack is the  combination of  technologies a company  uses to build and run an  application or project. It  helps us analyse and  understand various  technologies that needs to  be implemented in the  project. | 13 October 2022 |
| Data flow | A Data Flow Diagram  (DFD) is a traditional  visual representation of the information flows  within a system. A neat  and clear DFD can depict  the right amount of the  system requirement  graphically. It shows how  data enters and leaves the  system, what changes the  information, and where  data is stored. | 07 October 2022 |
| Sprint Delivery plan | Sprint Planning is an event  in scrum that defines what  can be delivered in the  upcoming sprint and how  that work will be achieved.  It helps us to organise and  complete the work  effectively and efficiently. | 20 October 2022 |
| Prepare milestone and  activity list | Helps us understand and  evaluate our progress and  accuracy so far. | 21 October 2022 |
| Project Development - Delivery of Sprint-1 | Develop and submit the  developed code by testing  it. | In progress |

**6.3 Reports from JIRA**

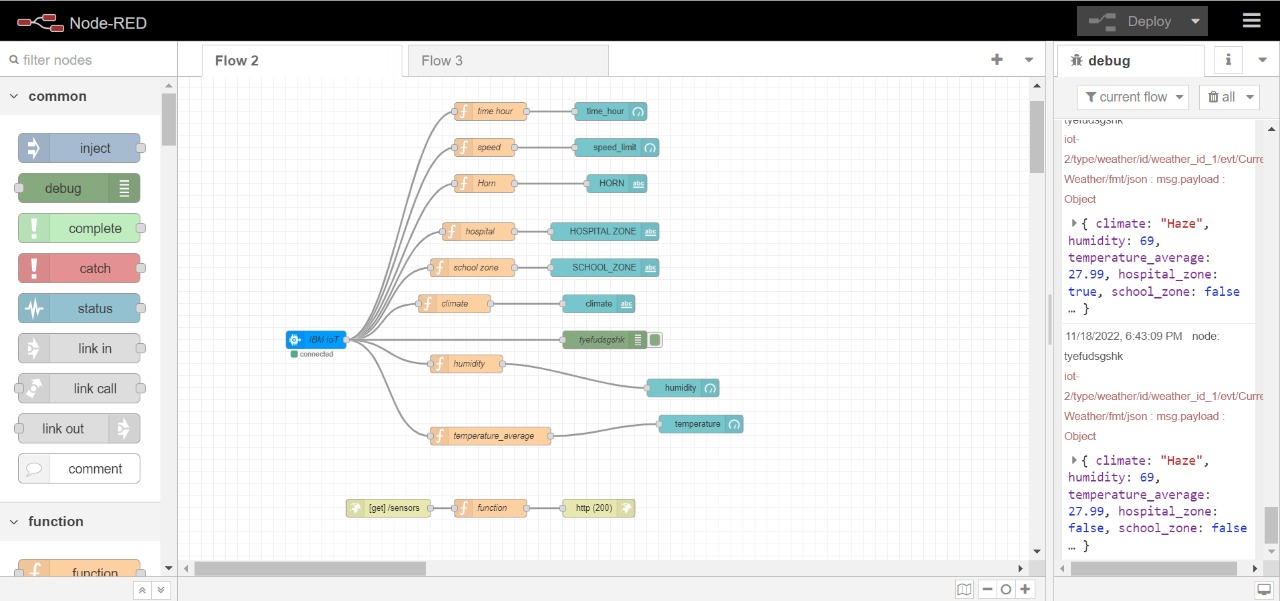




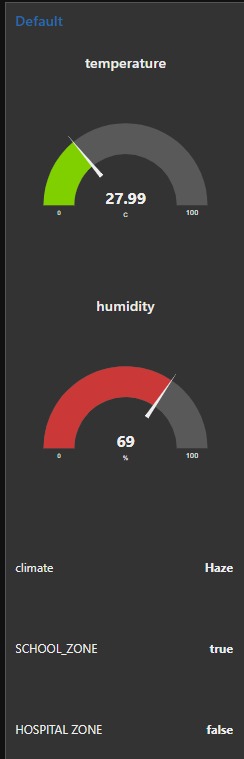
* 1. **CODING & SOLUTIONING**

**7.1 NODE RED**

Here, information is received from the IBM IoT Watson by using the node IBM IoT . From the IBM IoT node we get the function to display the objects. We had connected the dashboard's gauges and text to the function in order to display the output.

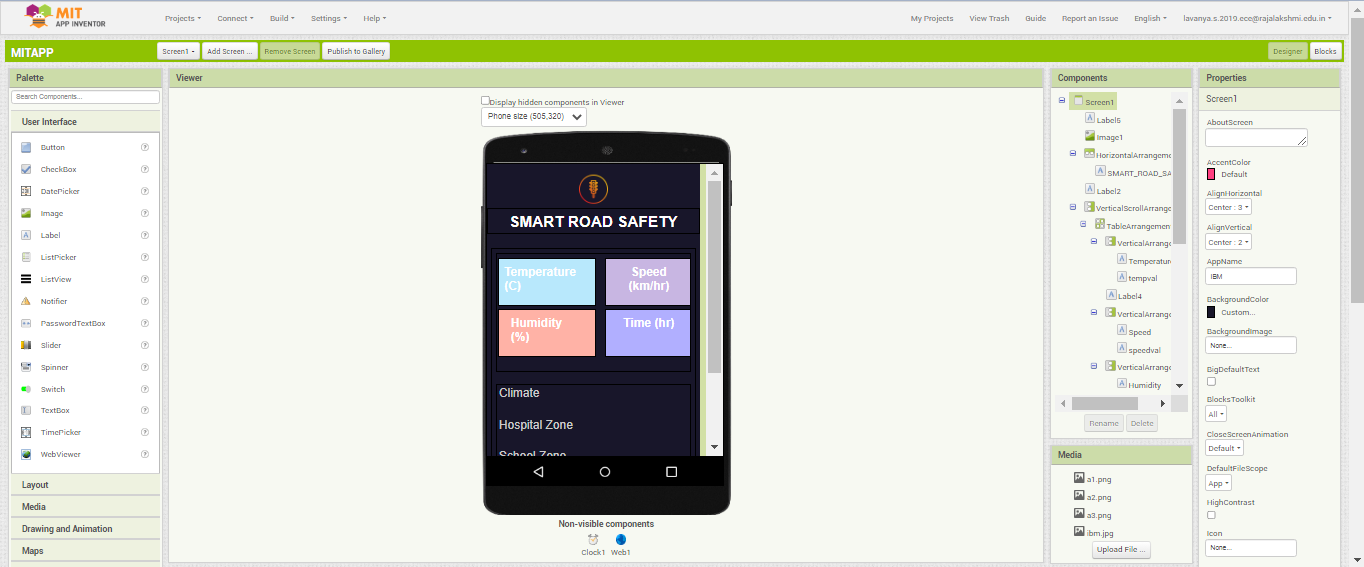
****

**Node Red Dashboard**

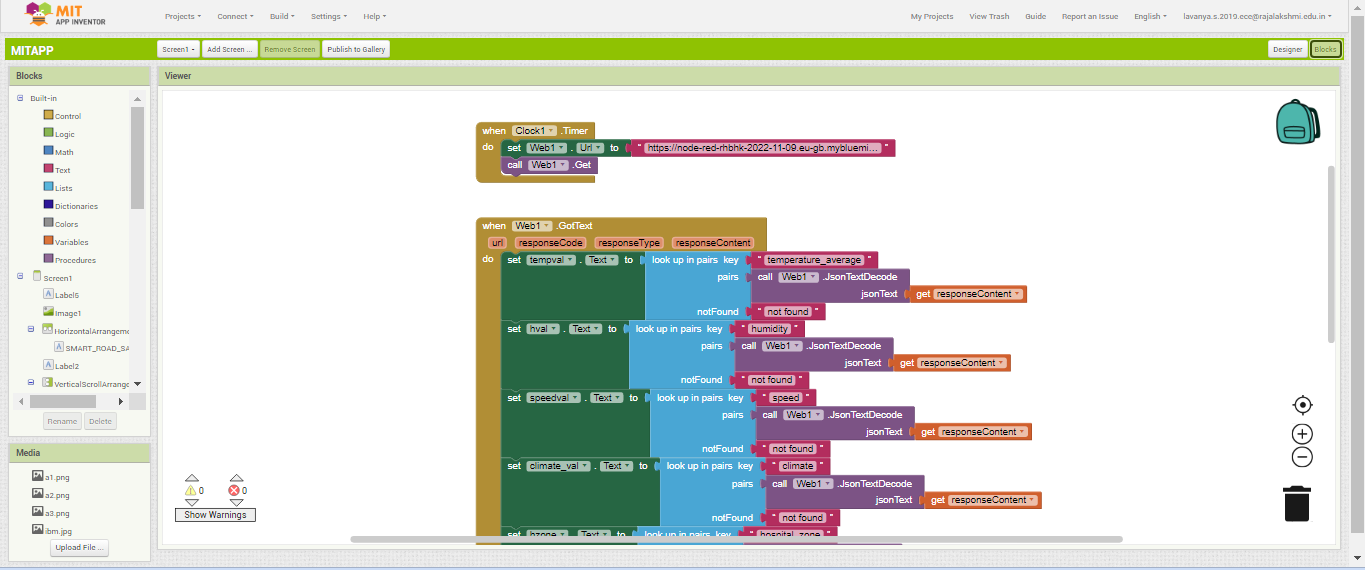
****

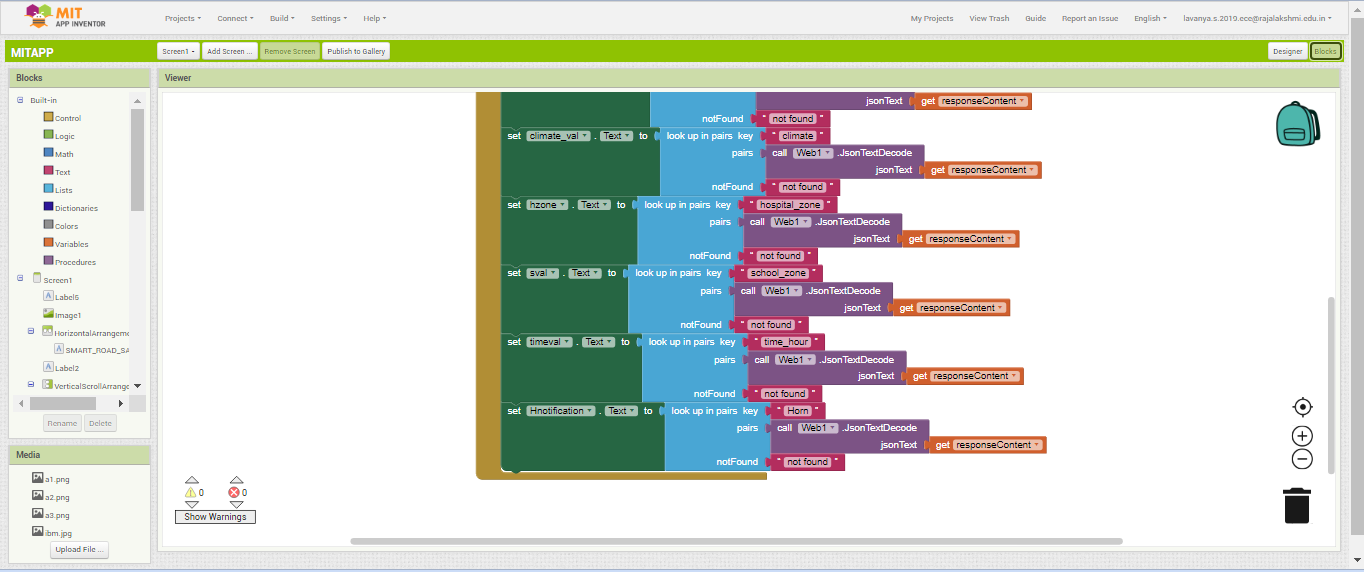
**7.2 User interface**

**MIT APP UI:**

****

**MIT APP Block Code:**

****

****

* 1. **TESTING**

**8.1 Test Cases**

**● TEST CASE 1**

Clear weather - Usual Speed Limit.(60 km/h)

**● TEST CASE 2**

Snow/Fog – Speed limkit is reduced to 10 km/h.

**● TEST CASE 3**

Rainy Weather – Set speed limit to 15 km/h.

**● TEST CASE 4**

School/Hosipital Zone – Usage of horn not allowed.

**8.2 User Acceptance Testing**

Dynamic speed & divertion variations based on the weather and traffic helps user to avoid traffic and have a safe journey home. The users would welcome this idea to be implemented everywhere.

* 1. **RESULTS**

**9.1 Performance Metrics**

Based on the IBM pack we chose, the performance of the website varies. Built upon NodeJS, a light and high performance engine, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, a even higher demand of customers can be served.

* 1. **ADVANTAGES & DISADVANTAGES**

**10.1 ADVANTAGES**

* The system is automatically updated based on current the weather condition
* maintenance cost is minimum
* safety is insured
* hassle free
* Posts warning signs
* Lower battery consumption and fuel consumption
* High processing speed
* Efficient
* Can enhance security measures on roads
* Reduce pollution levels
* Improve traffic flow

**10.2 DISADVANTAGES**

* Security issues
* Power consumption
  1. **CONCLUSION**

Our project is capable of serving as a replacement for static signs for a comparatively lower cost and can be implemented in the very near future. This will help reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

* 1. **FUTURE SCOPE**

Smart road technology includes a variety of sensors and analytics that improve transportation, optimize routes, and detect accidents. Overall, smart road technology has the potential to improve [transportation](https://tidd.ly/3tL8TCg) by optimizing routes and detecting accidents. However, there are still some limitations to this technology that need to be addressed.

* 1. **APPENDIX**

**Source Code**

#Node Red Link : https://node-red-rhbhk-2022-11-09.eu-gb.mybluemix.net/ui/#!/0?socketid=F2ttFCfJXR0NOxeQAAAx

import json

import os

import random as rd

import sys

import time

from datetime import datetime as dt

import ibmiotf.application

import ibmiotf.device

import requests as req

# Weather Details

api\_key = "fb12afe2efc8992de27da4156f22f05a"

city = "Chennai"

country = "IN"

url\_w=f"https://api.openweathermap.org/data/2.5/weather?q={city},{country}&appid={api\_key}&units=metric"

# Cloud Details for weather

org\_id\_w = "nbl97v"

device\_type\_w = "weather"

device\_id\_w = "weather\_id\_1"

auth\_method\_w = "token"

auth\_token\_w = "Kanish@2002"

# Datas Needed

default\_speed\_limit = 60 # km/hrs

default\_horn = True # Horn can be used

hour\_now = int(str(dt.now()).split()[1].split(":")[0])

today = str(dt.now().strftime("%A"))

# Location Information

location\_info = {

"school" : {

"school\_zone" : False, # Randomize Zones

"active\_time" : [7,17] # 7 am - 5 pm

}

,

"hospitals\_near\_by" :{

'hospital\_zone':False}, # Randomize Zones

"speed\_limit" : default\_speed\_limit,

"horn" : default\_horn

}

def get\_weather\_details():

weather\_req = req.get(url = url\_w)

weather\_data = weather\_req.json()

climate = weather\_data['weather'][0]['main']

humidity = weather\_data['main']['humidity']

pressure = weather\_data['main']['pressure']

temperature = weather\_data['main']['temp']

temperature\_min = weather\_data['main']['temp\_min']

temperature\_max = weather\_data['main']['temp\_max']

temperature\_feel = weather\_data['main']['feels\_like']

temperature\_average = (temperature\_max+ temperature\_min)/2

data = {'climate':climate,'humidity':humidity,'temperature\_average':temperature\_average}

return data

def myonpublishcallback\_w(data):

print(f"Published Temperature = { data['temperature\_average'] }, Humidity = {data['humidity']}, Climate = {data['climate']}")

def myonpublishcallback\_s(speed\_horn\_data):

print()

print(f"Speed limit = { speed\_horn\_data['speed'] }")

print(f"Horn Info = {speed\_horn\_data['Horn']}")

print(f"hospital\_zone = {speed\_horn\_data['hospital\_zone']}")

print(f"school\_zone = {speed\_horn\_data['school\_zone']}")

print(f"Time Now (in hrs) = {speed\_horn\_data['time\_hour']}")

print()

# Speed Limit and Horn Process

def speed\_process(climate):

#print(climatee)

#print(location\_info)

if climate == 'Rain':

if location\_info['hospitals\_near\_by']['hospital\_zone']:

location\_info['horn'] = False

location\_info['speed\_limit'] = 15

elif location\_info['school']['school\_zone']:

if today == "Sunday":

location\_info['horn'] = default\_horn

location\_info['speed\_limit'] = 25

else:

if location\_info['school']['active\_time'][0] >= hour\_now and location\_info['school']['active\_time'][1]<=hour\_now :

location\_info['horn'] = False

location\_info['speed\_limit'] = 15

else:

location\_info['horn'] = default\_horn

location\_info['speed\_limit'] = 25

else:

location\_info["horn"] = default\_horn

location\_info["speed\_limit"] = 20

elif climate == 'Snow' or climate == 'Smog' or climate == 'Fog':

location\_info['speed\_limit'] = 10

if location\_info['hospitals\_near\_by']['hospital\_zone']:

location\_info['horn'] = False

elif location\_info['school']['school\_zone']:

if today == "Sunday":

location\_info['horn'] = default\_horn

else:

if location\_info['school']['active\_time'][0] >=hour\_now and location\_info['school']['active\_time'][1]<=hour\_now :

location\_info['horn'] = False

else:

location\_info['horn'] = default\_horn

else:

location\_info["horn"] = default\_horn

else:

if location\_info['hospitals\_near\_by']['hospital\_zone']:

location\_info['horn'] = False

location\_info['speed\_limit'] = 20

elif location\_info['school']['school\_zone']:

if today == "Sunday":

location\_info['horn'] = default\_horn

location\_info['speed\_limit'] = default\_speed\_limit

else:

if (location\_info['school']['active\_time'][0] >=hour\_now ) and (location\_info['school']['active\_time'][1]<=hour\_now) :

location\_info['horn'] = False

location\_info['speed\_limit'] = 20

else:

location\_info['horn'] = default\_horn

location\_info['speed\_limit'] = default\_speed\_limit

else:

location\_info['horn'] = default\_horn

location\_info['speed\_limit'] = default\_speed\_limit

# Connecting to weather cloud

try:

deviceOptions\_w = {"org" : org\_id\_w,

"type" : device\_type\_w,

"id" : device\_id\_w,

"auth-method" : auth\_method\_w,

"auth-token" : auth\_token\_w

}

deviceCli\_w = ibmiotf.device.Client(deviceOptions\_w)

except Exception as e:

print(f"Caught exception connecting device {str(e)}")

sys.exit()

deviceCli\_w.connect()

# Processing

while True:

location\_info['hospitals\_near\_by']['hospital\_zone']=rd.choice([True,False])

location\_info['school']['school\_zone']=rd.choice([True,False])

climatee=rd.choice(["Rain","Fog","Mist","Smog","Snow"])

print(climatee)

data = get\_weather\_details()

speed\_process(climatee)

horn\_data = "Usage of Horn Allowed" if location\_info['horn'] else "Do not use the horn frequently"

speed\_horn\_data = {"hospital\_zone":location\_info['hospitals\_near\_by']['hospital\_zone'],"school\_zone":location\_info['school']['school\_zone'],"time\_hour":hour\_now,"speed":location\_info['speed\_limit'],"Horn":horn\_data}

for key in speed\_horn\_data:

data[key] = speed\_horn\_data[key]

success\_w = deviceCli\_w.publishEvent("Current Weather","json",data,qos=1,on\_publish = myonpublishcallback\_w(data))

if not success\_w :

time.sleep(1)

time.sleep(3)

**GitHub & Project Demo Link**

**GITHUB LINK:**

<https://github.com/IBM-EPBL/IBM-Project-20386-1659718378>

**Project Demo Link**

<https://github.com/IBM-EPBL/IBM-Project-20386-1659718378/blob/main/Final%20Deliverables/ibm_op.mkv>