Final Deliverables
Project Report

Signs with Smart Connectivity for Better Road Safety

Date	**19 November 2022**
Team ID	**PNT2022TMID41539**
Project Name	**Signs with Smart Connectivity for Better Road Safety**
Team Lead	**Mugila.R**
Team Menbers	**1. Ishwariya.P** **2. Kalpana.T** **3. Shreein Fathima.S**
College	**Idhaya Engineering College for Women – Chinnasalem** **Affiliated to Anna University**
Degree	**B.E. – Electronics and Communication Engineering**
Year/Semester	**Final year/7th Semester**

Contents

1. Introduction:

Roads are the foremost source of linking between cities and villages. Due to the ease of traveling by road, vehicles have become the main way people travel. The chances of vehicular accidents (Vas) have increased with the growing number of vehicles on the roads. During a journey, one does not know what will happen on the next road, particularly during bad weather conditions (BWC). In such a situation, driving can be difficult due to bad visibility, which can lead to an accident. It was also noticed that in BWC, multiple vehicle collisions (MVCs) can occur owing to delays in receiving information about an incident. According to one study by the Islamabad police, there were 9582 accidents from 2016 to 2017 all over Pakistan, involving 11,317 vehicles, leading to 5047 fatalities and 12,696 persons injured

Digital technologies like the Internet of Things (IoT) are reshaping road safety measures. Many technology initiatives are undertaken the world over to make smarter and safer roads, the ones that can interact with traffic and pedestrians. Assuming that by giving in vehicle technology information to the driver, accidents can be averted, several technology-based products have been developed. The latest technology researchers are working on is based on the Internet of Things (IoT). IoT is all about data. Data is becoming a valuable resource for our world.

Many sectors and industries have adopted IoT to reduce errors and improve performance in manufacturing, energy, health care, and communication. The WHO describes different measures that can be implemented with minimal economic impacts in its “Save LIVES: Road Safety Technical Package”. A cornerstone of these steps is realizing economic systems for “monitoring road safety by strengthening data systems”. Meanwhile, a key theme in the package is motivating the adoption of a Safe System approach, which is a holistic approach to road safety that parts from traditional management solutions by emphasizing safety by design.

Mobile-phone-based applications use built-in sensor data to detect the speed limit based on environmental situations.

The main contributions of this research are

1. Abriefsurveyonthestateoftheartrelatedtopre-accidentaswellaspost-accident models, frameworks, andtechniques;

2.Identificationandreportingoflimitationsinpreviousstudiesrelatedtoaccident detection;

3.The concept of a smart road with an event-sensing capability, plus implementation and testing through variousexperiments;

4. Demonstrationofanewandmodernwaytoquicklydetectaccident sand communicate with nearby vehicles andEOCs.

The risks for loss of life, injuries, and other damage may increase if an incident is not reported to an EOC in a timely fashion. Lives can be saved by sending timely information about an accident through an automated mechanism. Moreover, quick automobile accident detection and an alert system are required to protect approaching vehicles against an MVC. Several methods have been implemented in advanced vehicles (Avs) for avoiding an accident. An accident threat is detected through sensors installed in vehicles or by using smartphone sensors. Previous researchers have used accelerometers, smoke detectors, infrared (IR) obstacle sensors, proximity sensors, and biosensors to detect an accident.

1.1. Project Overview:

The main aim of this project is to help people automate the roads by providing them with a Web App through which they can monitor the parameters of the road like temperature, speed limit, and visibility of the road.They also show guides for schools and provide services of displaying hospitals, and restaurant signs accordingly.

1.2Purpose:

A large amount of research is being carried out in the domain of accident avoidance and accident alarms by a large number of researchers and practitioners. To avoid accidents, many approaches are utilized to enhance safety. For ease of reference, the literature on accident detection and avoidance is separated into three approaches: stand-alone, cooperative, and hybrid. Stand-alone approaches use sensors, such as radar and light detection and ranging (LiDAR), for accident avoidance and detection, whereas cooperative approaches rely on V2X technology and hybrid approaches.

2. LITERATURE SURVEY:

Abstract:

In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases.We can consider some cases when there are road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system that has digital signboards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data on road diversions, accident-prone areas, and information sign boards can be entered through the web app. This data is retrieved and displayed on the signboards accordingly.

Introduction:

An automated deep learning (DL)-based system was developed for detecting accidents from video data. The system uses visual components in temporal order to represent traffic collisions. As a result, the model architecture is composed of a visual-features-extraction phase followed by transient pattern identification. Convolution and recurrent layers are used in the training phase to learn visual and temporal features. In public traffic accident datasets, an accuracy of 98% was attained in the detection of accidents, demonstrating a strong capacity for detection independent of the road structure. The solution is limited to automobile crashes, not motorbikes, bicycles, and pedestrians. Furthermore, the model makes mistakes when determining accident segments under poor illumination (e.g., at night), at low resolutions, and when there are occlusions.

An accident management system was proposed in that makes use of cellular technology in public transportation. This method enables communication across various components, including those in ambulances, RSUs, and servers. Furthermore, in this system, an optimal route-planning algorithm (ORPA) is proposed to optimize aggregate spatial utilization of road networks while lowering the travel cost to operate a vehicle. The ORPA was evaluated through simulations, and findings were compared with other current algorithms. In congested areas, the proposed method can also be used to offer fast routes for ambulances. All vehicles, including ambulances, are required to have a route indicator installed, as well as the ability to use remote correspondence. The ORPA outperformed in terms of average speed and travel duration, according to the evaluation data. The proposed system only works for predicted patterns and can fail due to the unpredicted behavior of traff

Existingproblem:

The Safe System Approach

The Safe System (SS) approach to transport networks originated with the “Safe Road Transport System” model developed by the Swedish Transport Agency. In its essence, the approach migrates from the view that accidents are largely and

automatically the driver's fault to a view that identifies and evaluates the true causes of accidents. Through the categorization of safety into the safety of three elements (vehicle, road, and road user), SS minimizes fatalities and injuries by controlling speeds and facilitating prompt emergency response. The model has been widely adopted since its introduction and is currently motivated by the WHO as a basis for road safety planning, policy-making, and enforcement

REFERENCE:

1. World Health Organization, "Global status report on road safety 2015,"https://www.who.int/violence_injury_prevention/road_safety_status/2015/en/. View at: GoogleScholar
2. World Health Organization, "Decade ofAction for Road Safety 2011-2020 seeks to save millions of lives," http://www.who.int/roadsafety/decade_of_action/en/. View at: Publisher Site | Google Scholar
3. F. Wegman, "The future of road safety: A worldwide perspective," *IATSS Research*, vol. 40, no. 2, pp. 66–71,2017. View at: Publisher Site | Google Scholar
4. World HealthOrganization,"SaveLIVES-Aroadsafetytechnical package,"2017. View at: Google Scholar
5. W. E. Marshall, "Understanding international road safety disparities: Why is Australia so much safer than the United States?" *Accident Analysis & Prevention*, vol. 111, pp. 251–265,2018. View at: Publisher Site | Google Scholar
6. "Open Street Maps, with New York Countyhighlighted," https://www.openstreetmap.org/relation/2552485. View at: Google Scholar

7. United States Census Bureau, “TIGER/Line® Shapefiles: Roads,”https://www.census.gov/cgi-bin/geo/shapefiles/index.php?year=2018&layergroup=Roads.
View at: Google Scholar
8. X. Wang, X. Wu, M. Abdel-Aty, and P. J. Tremont, “Investigation of road network features and safety performance,” *Accident Analysis & Prevention*, vol. 56, pp. 22–31,2013.
View at: Publisher Site | Google Scholar
9. EuropeanRoadAssessmentProgram(EuroRAP),“EuropeanRoadSafety Atlas,”http://atlas.eurorap.org/.
View at: Google Scholar
10. EuropeanRoadAssessmentProgramme(EuroRAP),“StarRatings,” http://www.eurorap.org/protocols/star-ratings/.
View at: Google Scholar
11. International Road Assessment Programme (iRAP),“iRAP,”https://www.irap.org/.
View at: Google Scholar
12. H.M.HassanandH.Al-Faleh,“Exploringtheriskfactorsassociatedwith the size and severity of roadway crash in Riyadh,” *Journal of Safety Research*, vol. 47, pp. 67–74,2013.
View at: Publisher Site | Google Scholar
13. E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, “Internet-of-things-based smart environments: State of theart, taxonomy, and open research challenges,” *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 10–16, 2016.
View at: Publisher Site | GoogleScholar
14. Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani,

"Internet-of-Things-BasedSmartCities:RecentAdvancesandChallenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 16–24,2017. View at: Publisher Site | Google Scholar

15. AARON,"GPSLoggerincludingGyro/Tilt/Compass&Accelerometer," https://www.aaronia.com/products/spectrum-analyzers/gps-logger/. View at: Google Scholar
16. M. Farsi, K. Ratcliff, and M. Barbosa, "Overview of controller area network," *Computing and Control Engineering Journal*, vol. 10, no. 3, pp. 113–120, 1999. View at: Publisher Site | GoogleScholar
17. MUNIC (company website),https://www.munic.io/.
18. Torque Pro (OBD 2 & Car), "Torque Pro (OBD2 & Car) Google Play Page," http://goo.gl/CWD5VT. View at: Google Scholar
19. J.-S. Zhou, S.-H.Chen, W.-D.Tsay, and M.-C. Lai, "The implementation of OBD-II vehicle diagnosis system integrated with cloud computation technology," in *Proceedings of the 2013 2nd International Conference on Robot, Vision and Signal Processing, RVSP 2013*, pp. 9–12, Japan, December 2013. View at: Google Scholar
20. A. Mednis, A. Elsts, and L. Selavo, "Embedded solution for road condition monitoringusingvehicularsensornetworks,"in*Proceedingsofthe20126th InternationalConferenceonApplicationofInformationandCommunication Technologies, AICT 2012*, Georgia, October2012. View at: Google Scholar
21. M.Yamada, K. Ueda, I. Horiba, and N. Sugie, "Discrimination of the Road Condition Toward Understanding of Vehicle Driving Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 1, pp. 26–31,2001.

View at: Publisher Site | Google Scholar

22. J.Eriksson,L.Girod,B.Hull,R.Newton,S.Madden,andH.Balakrishnan, "The pothole patrol: using a mobile sensor network for roadsurfacemonitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 29–39, Breckenridge, Colo, USA, June 2008.

View at: Publisher Site | Google Scholar

23. M. Valentino, C. Quiligotti, and L. Carone, "Branchial cleft cyst," *Journal of Ultrasound*, vol. 16, no. 1, pp. 17–20,2013.

View at: Publisher Site | Google Scholar

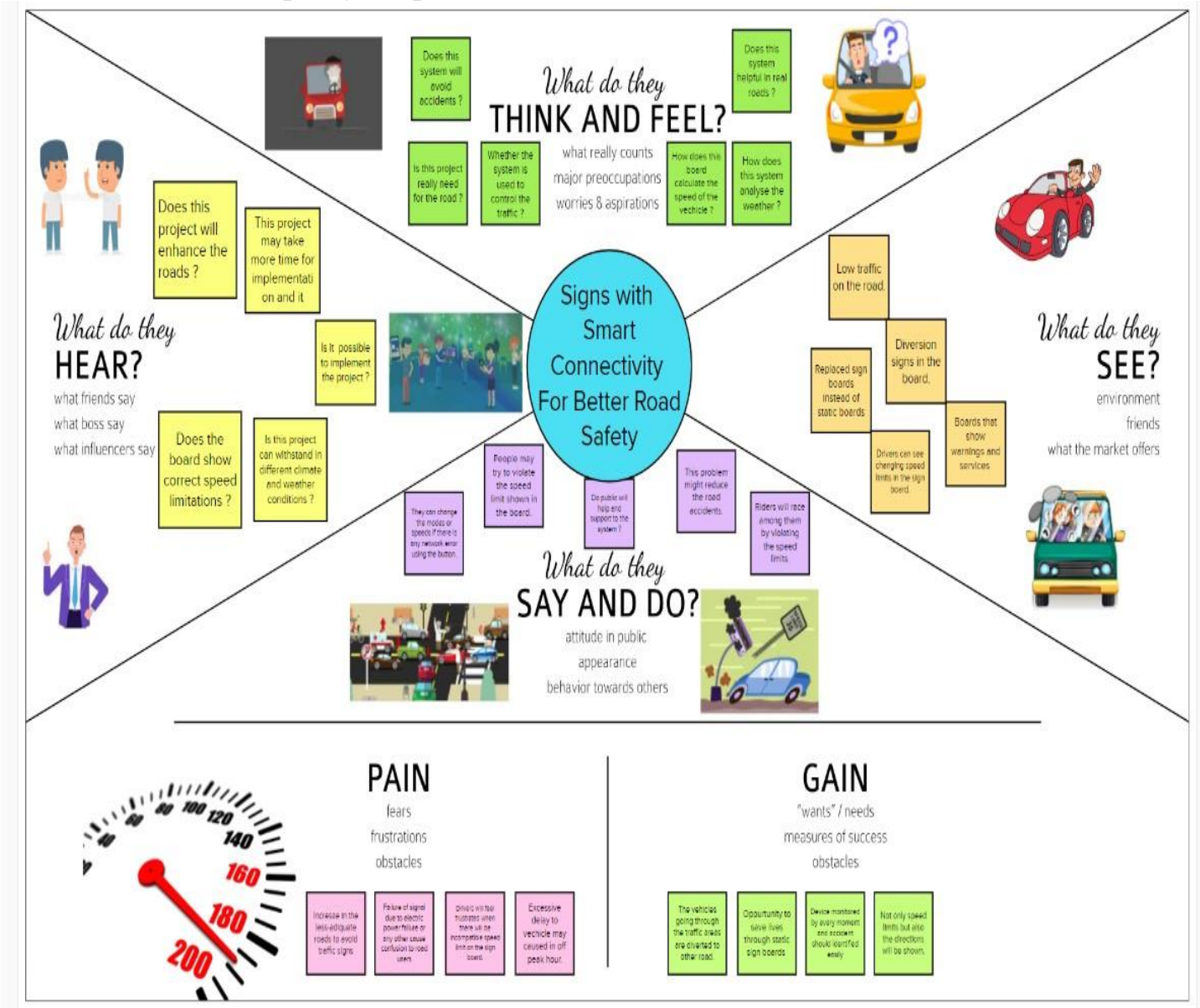
Problem StatementDefinition

A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the gap between the current (problem) state and desired (goal) state of a process or product. Focusing on the facts, the problem statement should be designed to address the Five Ws. The first condition of solving a problem is understanding the problem, which can be done by way of a problem statement.

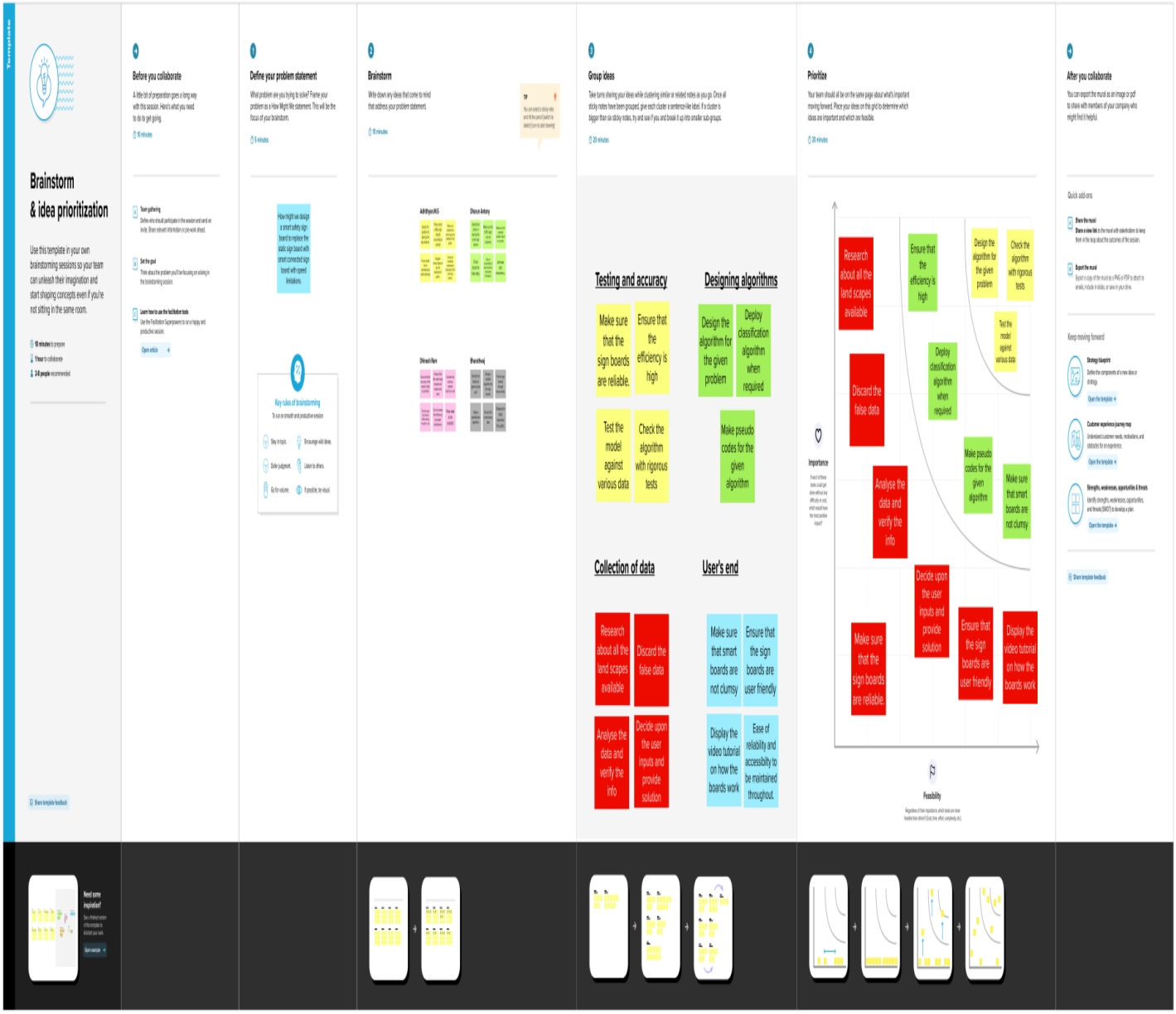
In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system that has digital signboards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data on road diversions, accident-prone areas, and information sign boards can be entered through the web app. This data is retrieved and displayed on the signboardsaccordingly.

3. IDEATION & PROPOSED SOLUTION:

Empathy MapCanvas:



Ideation &Brainstorming:



Proposed Solution:

The project team shall fill in the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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 / Solution 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 / Uniqueness	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.	Social Impact / 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 (Revenue 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.

Problem Solutionfit:

4. REQUIREMENTANALYSIS

Functionalrequirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Visibility**	Sign Boards should be made with LED's which are bright colored and are capable of attracting the drivers attention but it should also not be too distracting or blinding cause it may lead to accidents.
FR-2	**User Need**	The smart sign boards should be placed frequently in places it is needed and less in places where it is not needed much to avoid confusion for the user during travel.
FR-3	**User Understanding**	For better understanding of the driver, the signs should be big, clear and legible and it can also include illustrations which will make it easily understandable to the driver.
FR-4	**User Convenience**	The display should be big enough that it should even be visible from far distance clearly.

Non-Functionalrequirements:

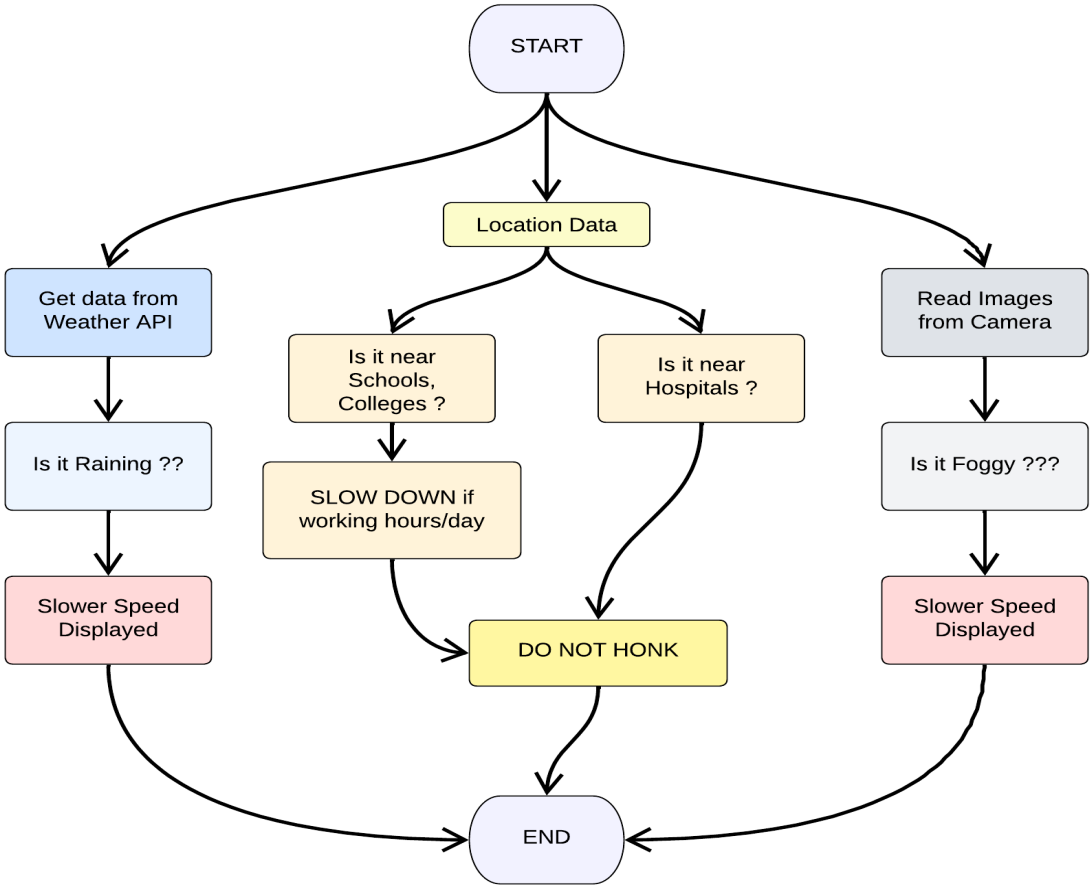
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It should be able to Upgrade and Update when there is a need for it.
NFR-2	**Security**	It should have good security system so that no other person is able to hack and display their own directions.
NFR-3	**Reliability**	It should be able to display to information correctly and error-free.
NFR-4	**Performance**	It should be able to automatically update itself when certain weather or traffic problem occurs.
NFR-5	**Availability**	It should be available 24/7 so that it can be beneficial to the customer i.e the driver.
NFR-6	**Scalability**	It should be able to easily change and upgrade according to change and need in requirement

5. PROJECTDESIGN:

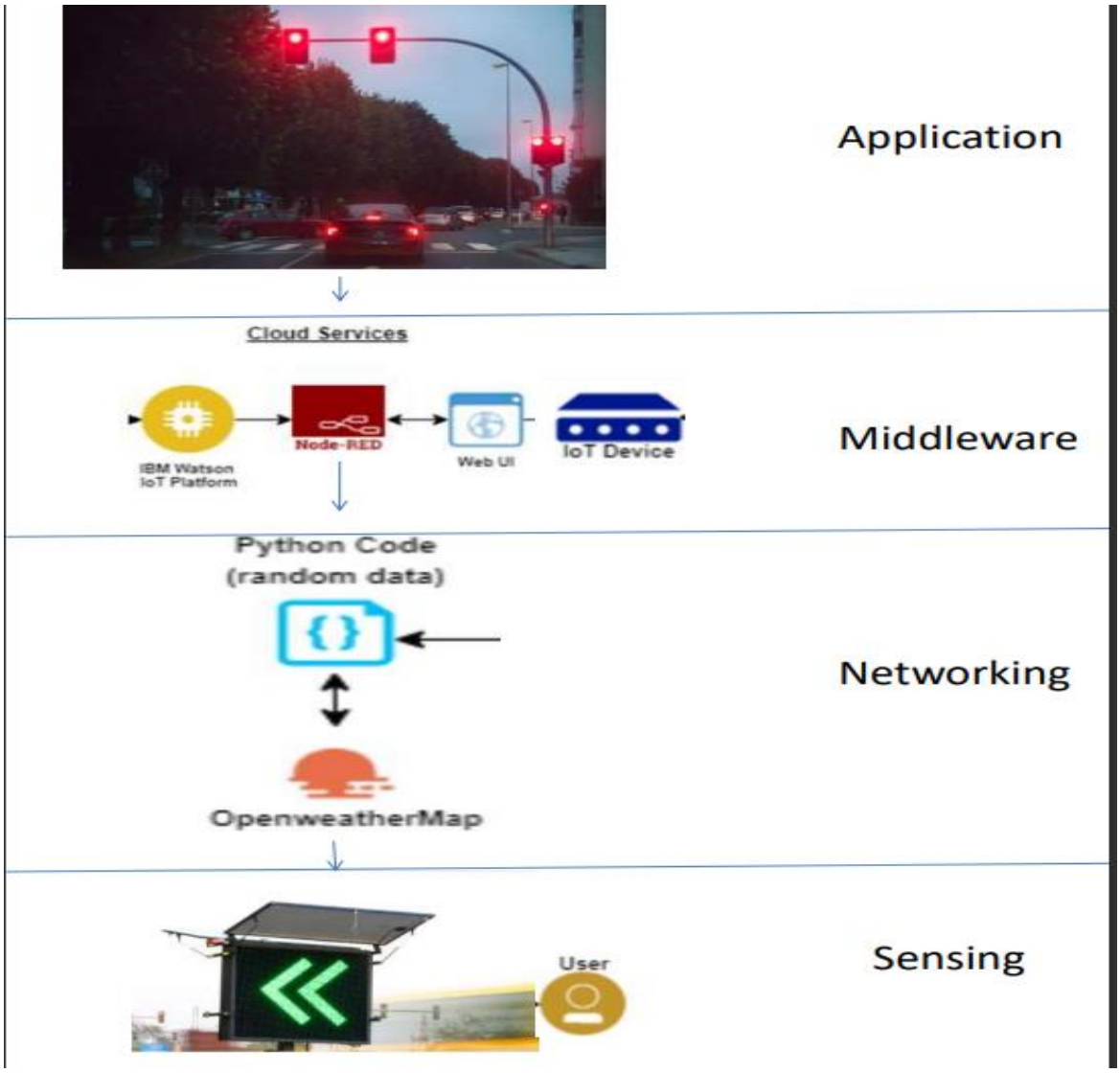
Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

Data FlowDiagrams:

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.



Solution &TechnicalArchitecture:



Userstories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	**UserStory/Task**	**Acceptancecriteria**	Priority	Release
Customer (Mobile user)	Registration	USN-1	**I can get my speed limitation using weather application .**	**I can receive speed limitations**	High	Sprint-1
		USN-2	**As a user, I can register for the application by entering my email, password, and confirming my password. As a user,**	**I can access my account / dashboard**	Medium	Sprint-2
		USN-3	**As a user, I can increase or decrease my speed according to the weather change**	**I can increase or decrease my speed**	High	Sprint-1
		USN-4	**As a user, I can I get my traffic diversion signs depending on the traffic and the fatal situations.**	**I can access my traffic status ahead in my travel**	Medium	Sprint-1
	Login	USN-5	**As a user, I can log into the open weather map by entering email & password**	**I can access the application through my Gmail login**	High	Sprint-2
	Interface	USN-6	**As a user the interface should be simple and easily accessible**	**I can access the interface easily**	High	Sprint-1

Customer (Web user)	Data generation	USN-7	**As a user I use open weather application to access the data regarding the weather changes.**	**I can access the data regarding the weather through the application**	High	Sprint-1
Administrator (Officials)	Problem solving/ Fault clearance	USN-8	**As an official who is in charge for the proper functioning of the sign boards have to maintain it through periodic monitoring.**	**Officials can monitor the sign boards for proper functioning.**	Medium	Sprint-2
			entering my email, password, and confirming my password	**dashboard**		

6. PROJECT PLANNING &SCHEDULING

The definition of a sprint is a dedicated period in which a set amount of work will be completed on a project. It's part of the agile methodology, and an Agile projectwill be broken down into a number of sprints, each sprint taking the project closer to completion**.**

Sprint Planning &Estimation:

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
Sprint-1	Preparation	USN-1	**Install the Python IDE.** Install the required python libraries: • Install Watson IoT Python SDK to connect to IBM Watson IoT Platform using python code: give the following command in command prompt: pip install wiotp-sdk Download the required files from the link Create a fast SMS service for sending the messages and getting the API Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/Software.pdf	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-2	**Create An Account In OpenweatherMap Website** Using Openweathermap we can get current weather details of a location and integrate this with our project Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/Create%20An%20Account%20In%20OpenweatherMap%20Website.pdf **Create An Account In MIT App inventor Website** Create an account in MIT app inventor website and download MIT AI2 companion app in mobile.	5	Low	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-3	**IBM Cloud Services** Need to have basic knowledge of the following cloud services: • IBM Watson IoT Platform	5	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
			• Node-RED Service Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/IBM%20Cloud%20Services.pdf Sprint-1 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-1/Sprint-1.pdf			
Sprint-2	Create And Configure IBM Cloud Services	USN-4	**Create IBM Watson IoT Platform And Device** • IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform. • In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials. • Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Create%20And%20Configure%20IBM%20Cloud%20Services/Create%20IBM%20Watson%20IoT%20Platform%20And%20Device.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-2	Create And Configure IBM Cloud Services	USN-5	**Create Node-RED Service** To create a web application create a Node-RED service. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Create%20And%20Configure%20IBM%20Cloud%20Services/Create%20Node-RED%20Service.pdf Sprint-2 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-2/Sprint-2.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
Sprint-3	Develop The Python Script	USN-6	**Develop A Python Script** Create a code snippet using python to • Extract weather data from OpenWeatherMap using APIs • Send the extracted data to the cloud • Receive data from the cloud and view it in the python compiler Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Develop%20The%20Python%20Script/Develop%20A%20Python%20Script.pdf	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-7	**Publish Data To The IBM Cloud** Python code is used to send random sensor data to the cloud and also to receive commands from the cloud. When the commands are received just print the statements which represent the control of the devices. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/tree/main/Develop%20The%20Python%20Script/Publish%20Data%20To%20The%20IBM%20Cloud Sprint-3 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-3/Sprint-3.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web Application Using Node-RED Service.	USN-8	**Develop The Web Application Using Node-RED** Configure the Node-RED flow to send data to the IBM IoT platform. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Develop%20A%20Web%20Application%20Using%20Node-RED%20Service/Develop%20The%20Web%20Application%20Using%20Node-RED/Create%20Node%20Red%20Flow%20To%20Send%20Data%20To%20Device.pdf	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web	USN-9	**Use Dashboard Nodes For Creating UI(Web App)**	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
	Application Using Node-RED Service.		Create use dashboard nodes to visualize the data in graphical format. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Develop%20A%20Web%20Application%20Using%20Node-RED%20Service/Use%20Dashboard%20Nodes%20For%20Creating%20UI(Web%20App)/UI%20Nodes%20Installation.pdf			4.Shreein Fathima S
Sprint-4	Develop A Mobile Application Using MIT App inventer.	USN-10	**Use MIT App For Creating Mobile application** Buidl the app. Using MIT AI2 companion(in mobile) by connecting it with the app builded(click connect in app inventor then click AI companion and scan the QR code in mobile companion- it will be connected) we can see the Road Safety informations in mobile. Sprint-4 link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-4/Sprint-4.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Preparation	USN-1	**Install the Python IDE.** Install the required python libraries: • Install Watson IoT Python SDK to connect to IBM Watson IoT Platform using python code: give the following command in command prompt: pip install	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			wiotp-sdk Download the required files from the link Create a fast SMS service for sending the messages and getting the API			
Sprint-1	Preparation	USN-2	**Create An Account In OpenweatherMap Website** Using Openweathermap we can get current weather details of a location and integrate this with our project **Create An Account In MIT App inventor Website** Create an account in MIT app inventor website and download MIT AI2 companion app in mobile.	5	Low	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-3	**IBM Cloud Services** Need to have basic knowledge of the following cloud services: • IBM Watson IoT Platform • Node-RED Service	5	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-2	Create And Configure IBM Cloud Services	USN-4	**Create IBM Watson IoT Platform And Device** • IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform. • In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			• Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.			
Sprint-2	Create And Configure IBM Cloud Services	USN-5	**Create Node-RED Service** To create a web application create a Node-RED service.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-6	**Develop A Python Script** Create a code snippet using python to • Extract weather data from OpenWeatherMap using APIs • Send the extracted data to the cloud • Receive data from the cloud and view it in the python compiler	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-7	**Publish Data To The IBM Cloud** Python code is used to send random sensor data to the cloud and also to receive commands from the cloud. When the commands are received just print the statements which represent the control of the devices.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web Application Using Node-RED Service.	USN-8	**Develop The Web Application Using Node-RED** Configure the Node-RED flow to send data to the IBM IoT platform.	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web	USN-9	**Use Dashboard Nodes For Creating UI(Web App)**	5	High	1.Mugila R 2.Ishwariya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Application Using Node-RED Service.		Create use dashboard nodes to visualize the data in graphical format.			P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Mobile Application Using MIT App inventer.	USN-10	**Use MIT App For Creating Mobile application** Build the app. Using MIT AI2 companion(in mobile) by connecting it with the app builded(click connect in app inventor then click AI companion and scan the QR code in mobile companion- it will be connected) we can see the Road Safety informations in mobile.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

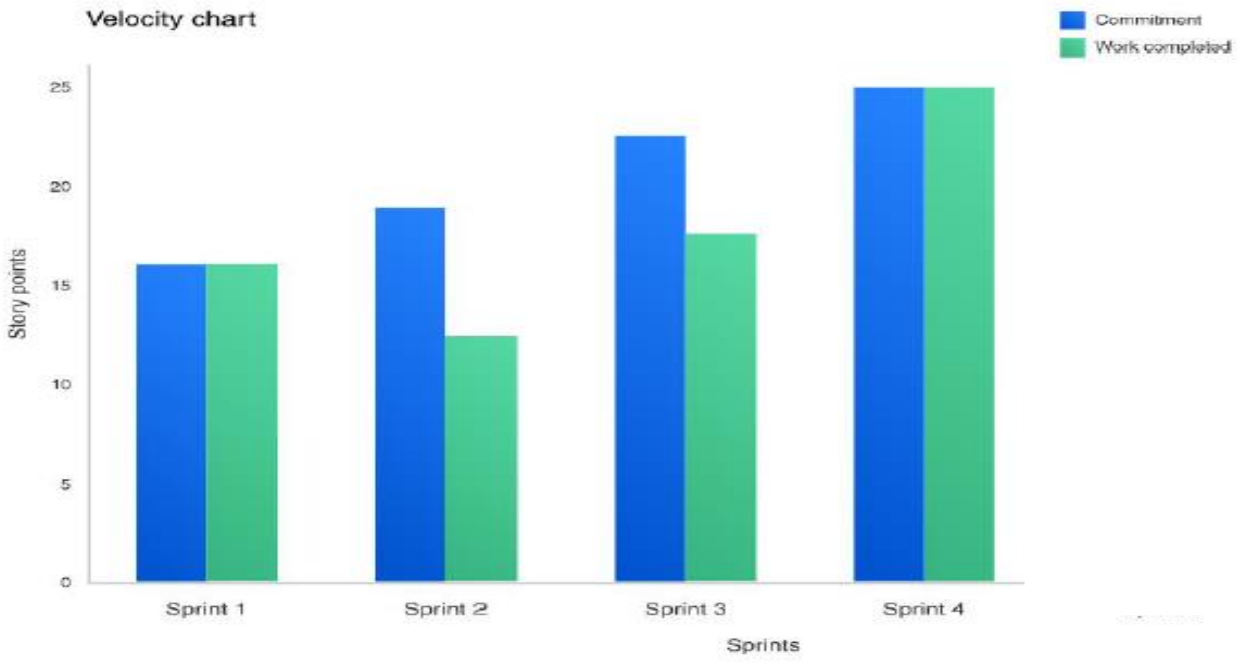
Velocity:

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 (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = 20/6 = 3.33$$

Burndown Chart:

A burn-down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as scrum. However, burn-down charts can be applied to any project containing measurable progress over time.



7.CODING & SOLUTIONING:

(Explain the features added in the project along with code):

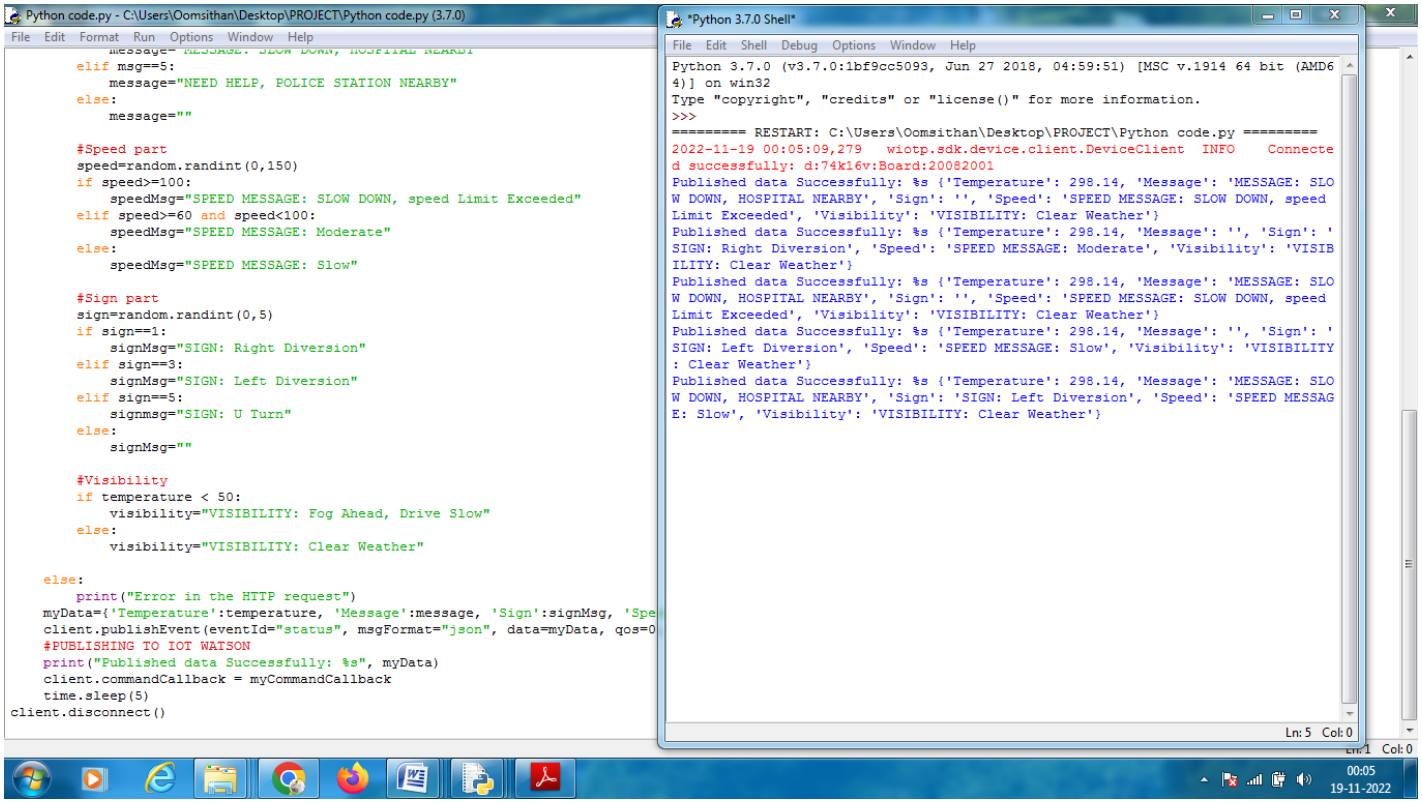
Feature 1 (coding andresult):

```
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = {
   #Configuration
   "identity": {
      "orgId": "3dpjnk",
      "typeId": "Sign_Board",
      "deviceId":"Board_1"
      },
   #API Key
   "auth": {
      "token": "1234567890"
   }
}

#Receiving callbacks from IBM IOT platform
def myCommandCallback(cmd):
```

```
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()

#OpenWeatherMap Credentials
BASE_URL ="https://api.openweathermap.org/data/2.5/weather?"
CITY = "Nagercoil"
URL = BASE_URL + "q=" + "Chennai" + "&appid=" + "01df65417ab3968e3fc2a38c4aee27bb"

while True:
    response = requests.get(URL)
    if response.status_code ==200:
        data = response.json()
        main = data['main']
        temperature =main['temp']
        humidity = main['humidity']
        pressure = main['pressure']
        report = data['visibility']

        #messge part
        msg=random.randint(0,5)
        if msg==1:
            message="SLOW DOWN, SCHOOL IS NEAR"
        elif msg==3:
            message="SLOW DOWN, HOSPITAL NEARBY"
        elif msg==5:
            message="NEED HELP, POLICE STATION NEARBY"
        else:
            message=""

        #Speed part
        speed=random.randint(0,150)
        if speed>=100:
            speedMsg=" SLOW DOWN, speed Limit Exceeded"
        elif speed>=60 and speed<100:
            speedMsg="Moderate"
        else:
            speedMsg="Slow"

        #Sign part
        sign=random.randint(0,5)
        if sign==1:
            signMsg="Right Diversion"
        elif sign==3:
            signMsg="Left Diversion"
        elif sign==5:
            signmsg="U Turn"
        else:
            signMsg=""

        #Visibility
        if temperature < 50:
```

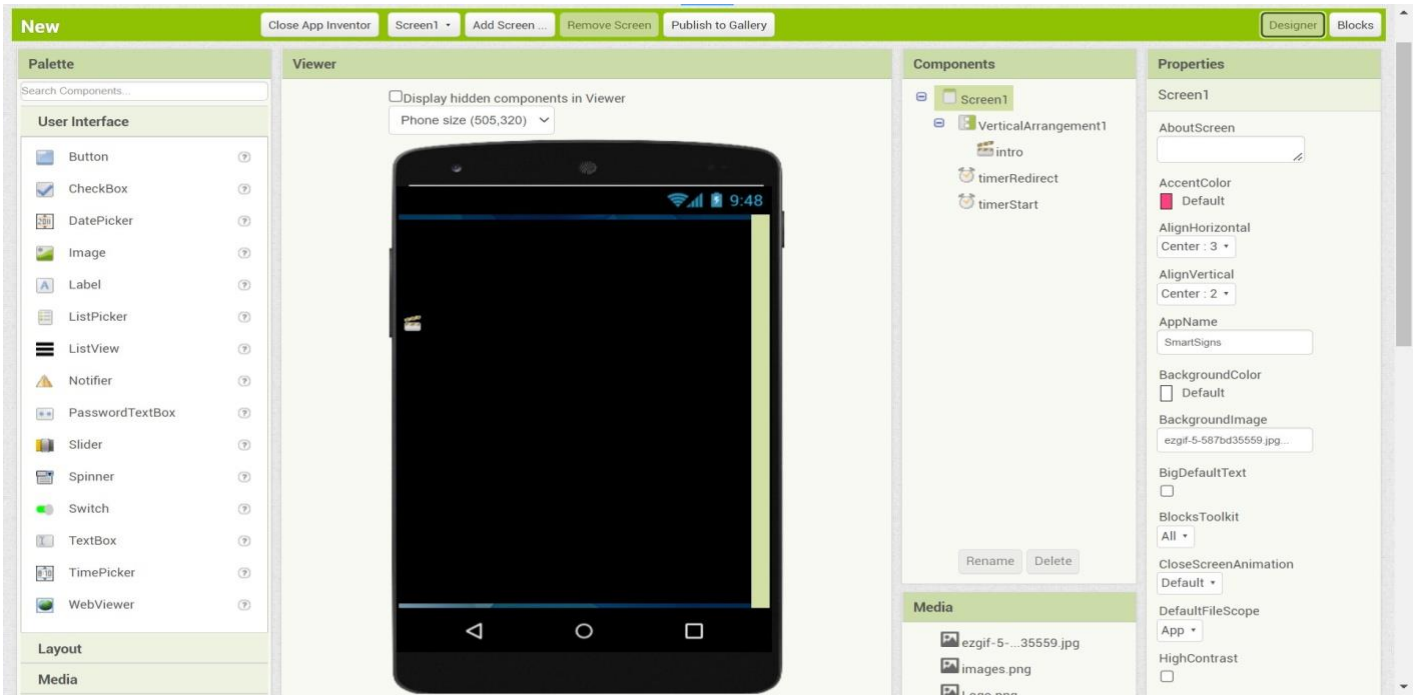
```
            visibility="Fog Ahead, Drive Slow"
        else:
            visibility="Clear Weather"

    else:
        print("Error in the HTTP request")
    myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg, 'Speed':speedMsg, 'Visibility':visibility}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    #PUBLISHING TO IOT WATSON
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

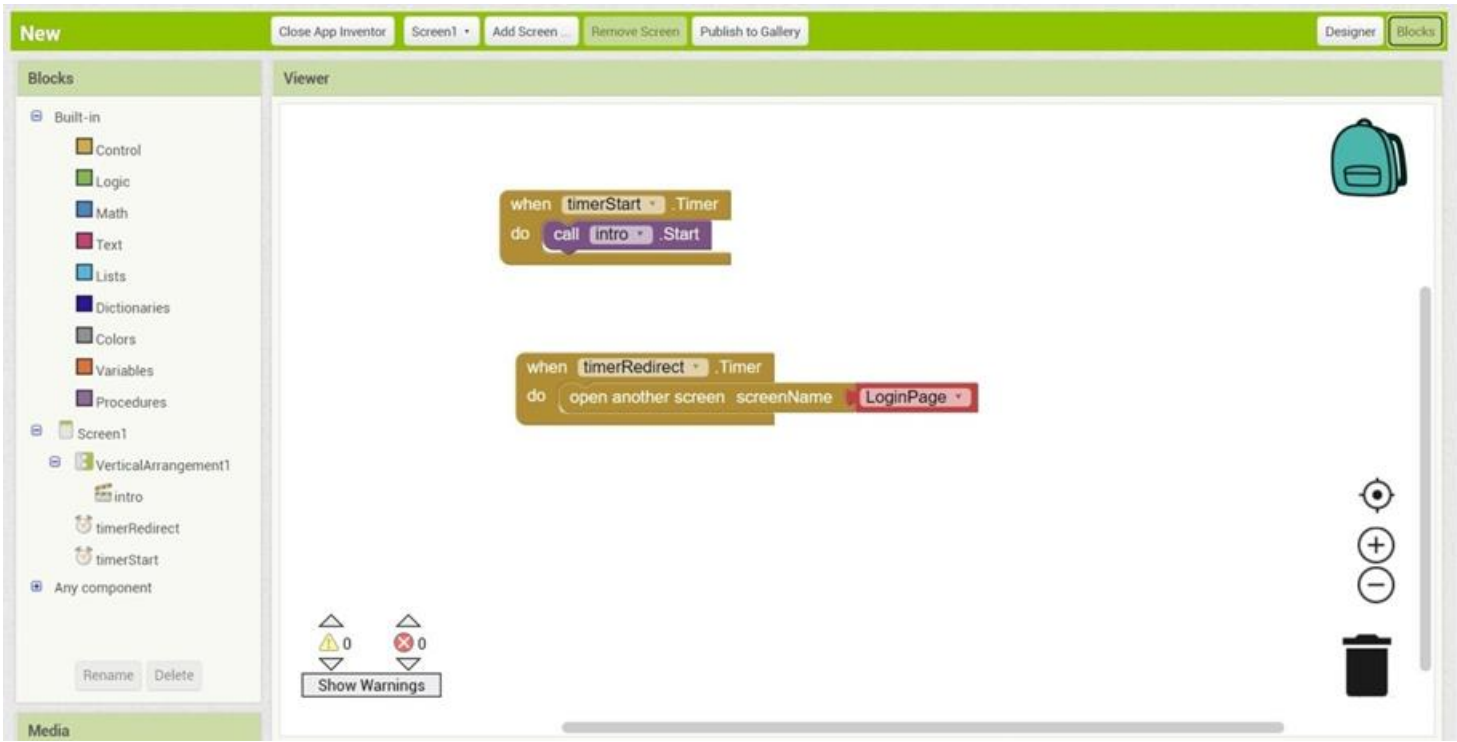
Output



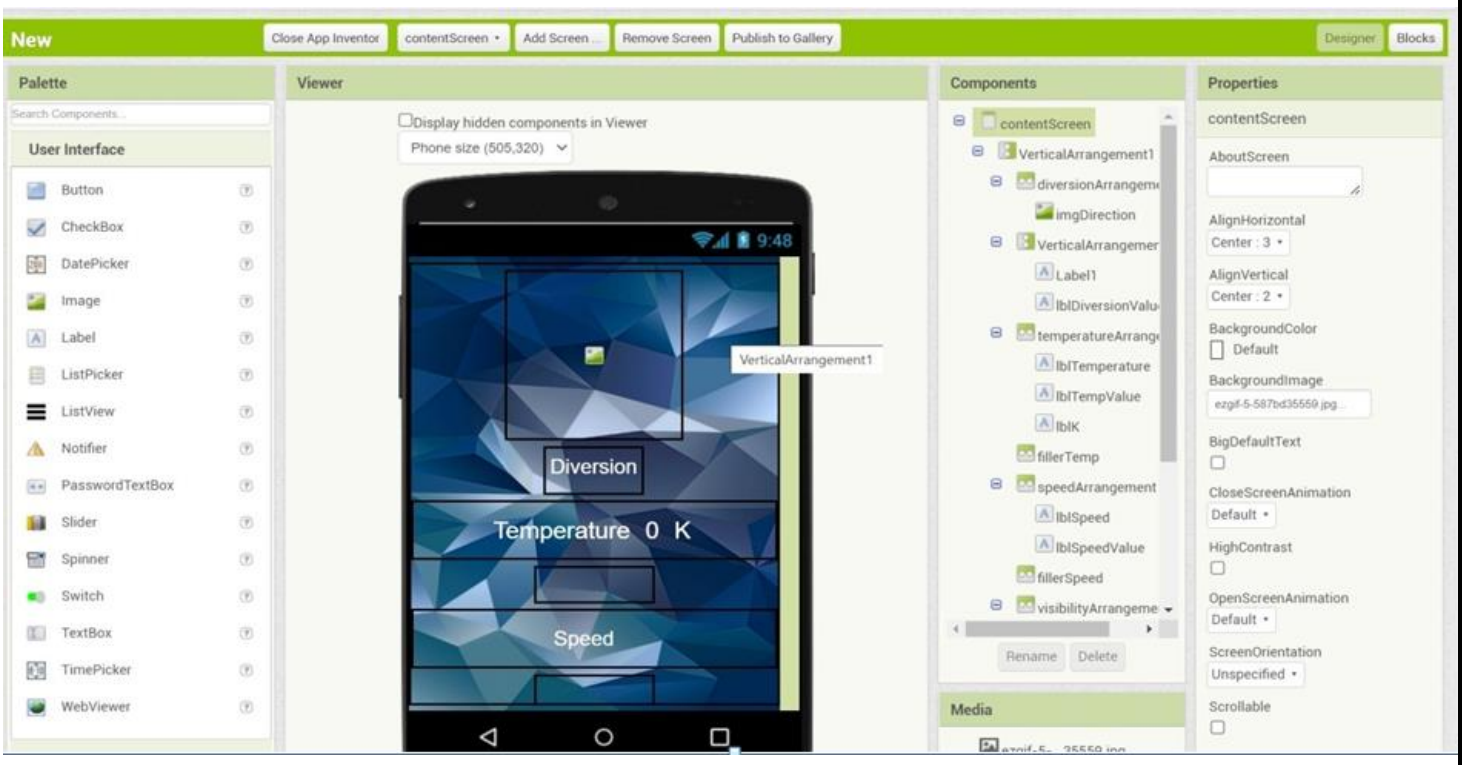
Feature 2: (MITAPPINVENTER):

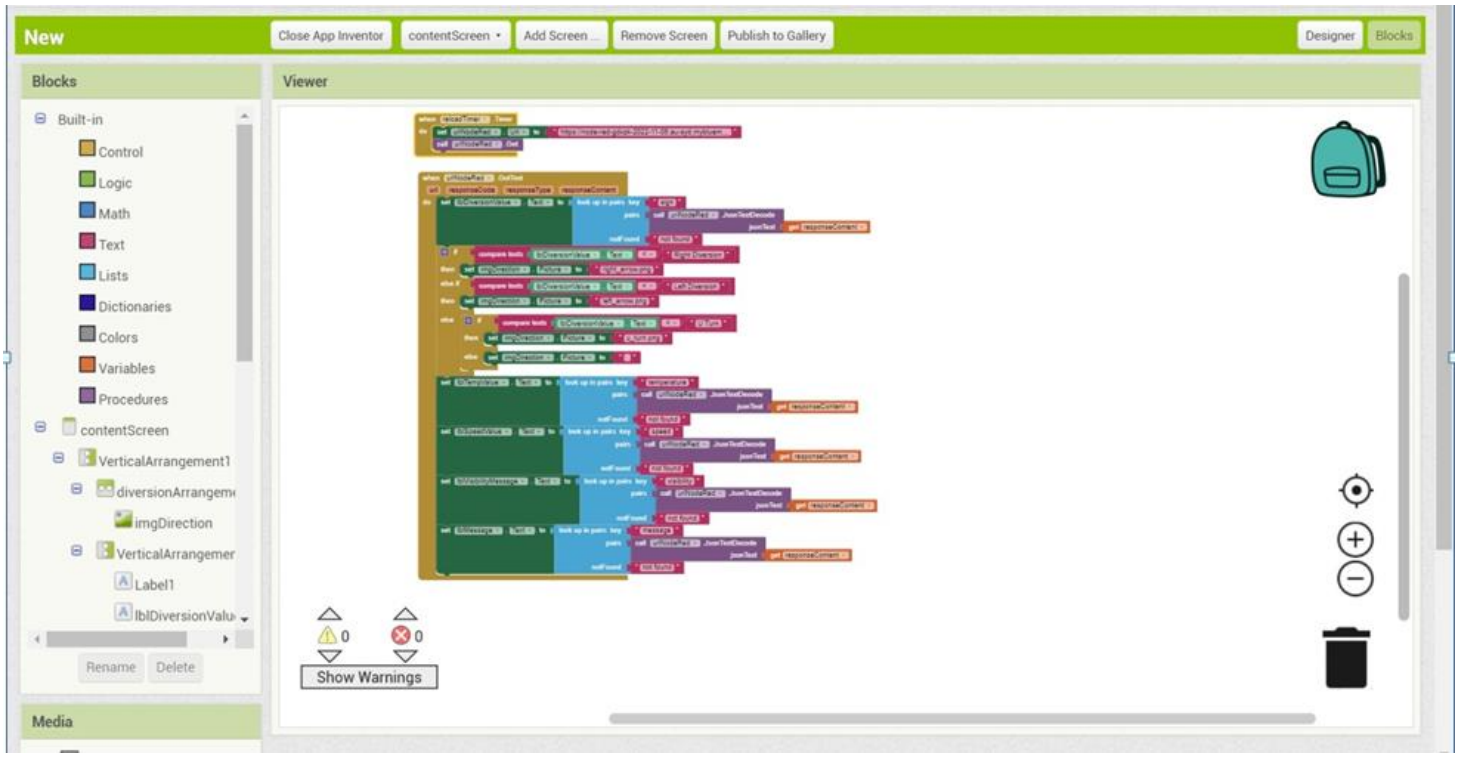
MIT APP INVENTOR: ICON PAGE:






New
Close App Inventor
LoginPage
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
LoginPage
loginPageArrangemen
loginArrangement
imgLogo
fillerAfterLogo
usernameArrangem
lblUsername
Rename
Delete
Media
Viewer
when btnLogin .Click
do if
txtUsername . Text = admin
and
txtPassword . Text = admin
then open another screen screenName contentScreen
else call checkCred .ShowAlert
notice Admin for both
0
0
Show Warnings


New
Close App Inventor
contentScreen
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
VerticalArrangement1
Diversion
Temperature 0 K
Speed
Components
contentScreen
VerticalArrangement1
diversionArrangem
imgDirection
VerticalArrangemer
Label1
lblDiversionValu
temperatureArrang
lblTemperature
lblTempValue
lblK
fillerTemp
speedArrangement
lblSpeed
lblSpeedValue
fillerSpeed
visibilityArrangeme
Rename
Delete
Media
Properties
contentScreen
AboutScreen
AlignHorizontal
Center : 3
AlignVertical
Center : 2
BackgroundColor
Default
BackgroundImage
ezgif-5-587bd35559.jpg...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable

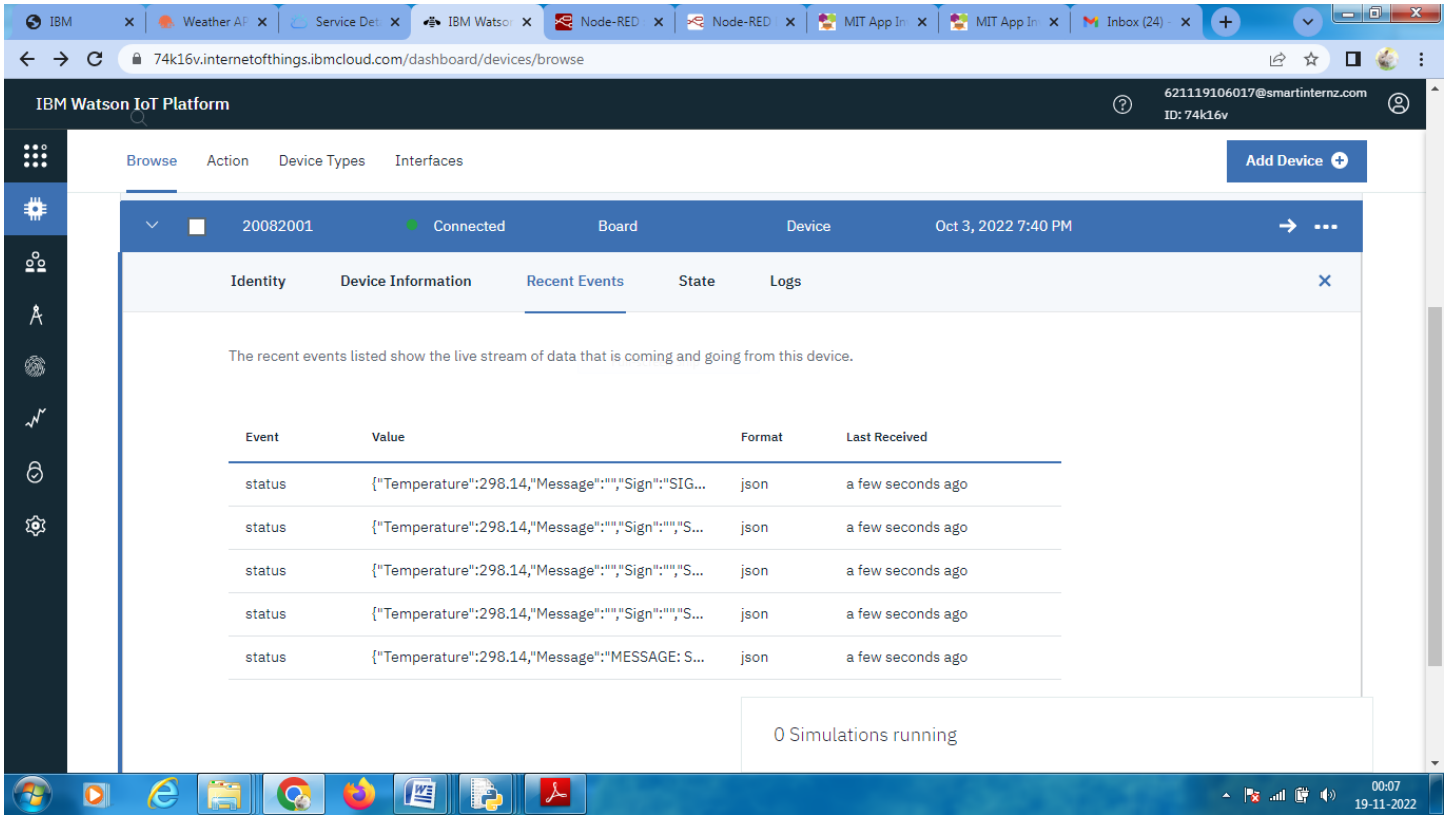

New
Close App Inventor
contentScreen
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
contentScreen
VerticalArrangement1
imgDirection
Label1
Rename
Delete
Media
Viewer
Show Warnings


Diversion
Temperature 26.36 °C
Speed Limit Exceeded
Visibility Clear Weather
SLOW DOWN, SCHOOL IS NEAR

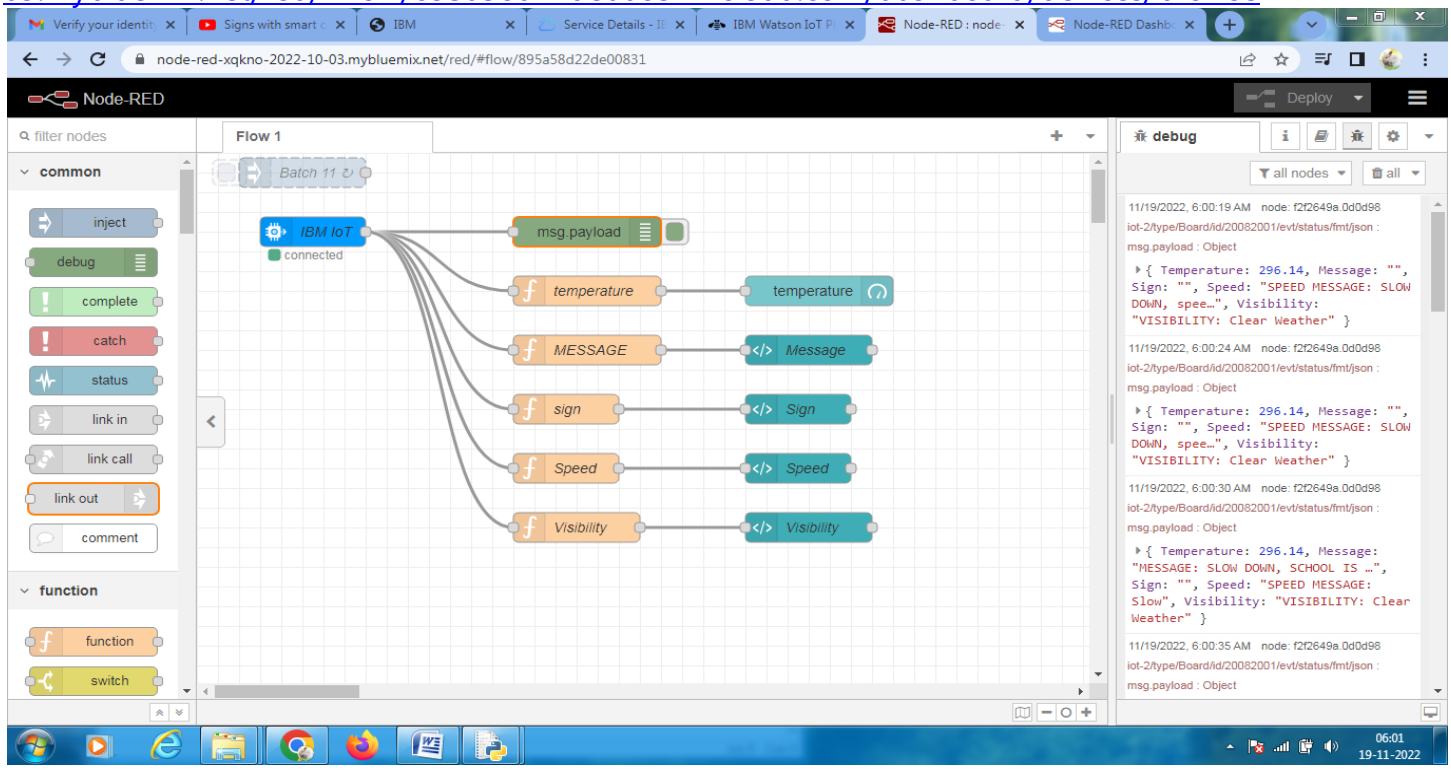
8.TESTING:

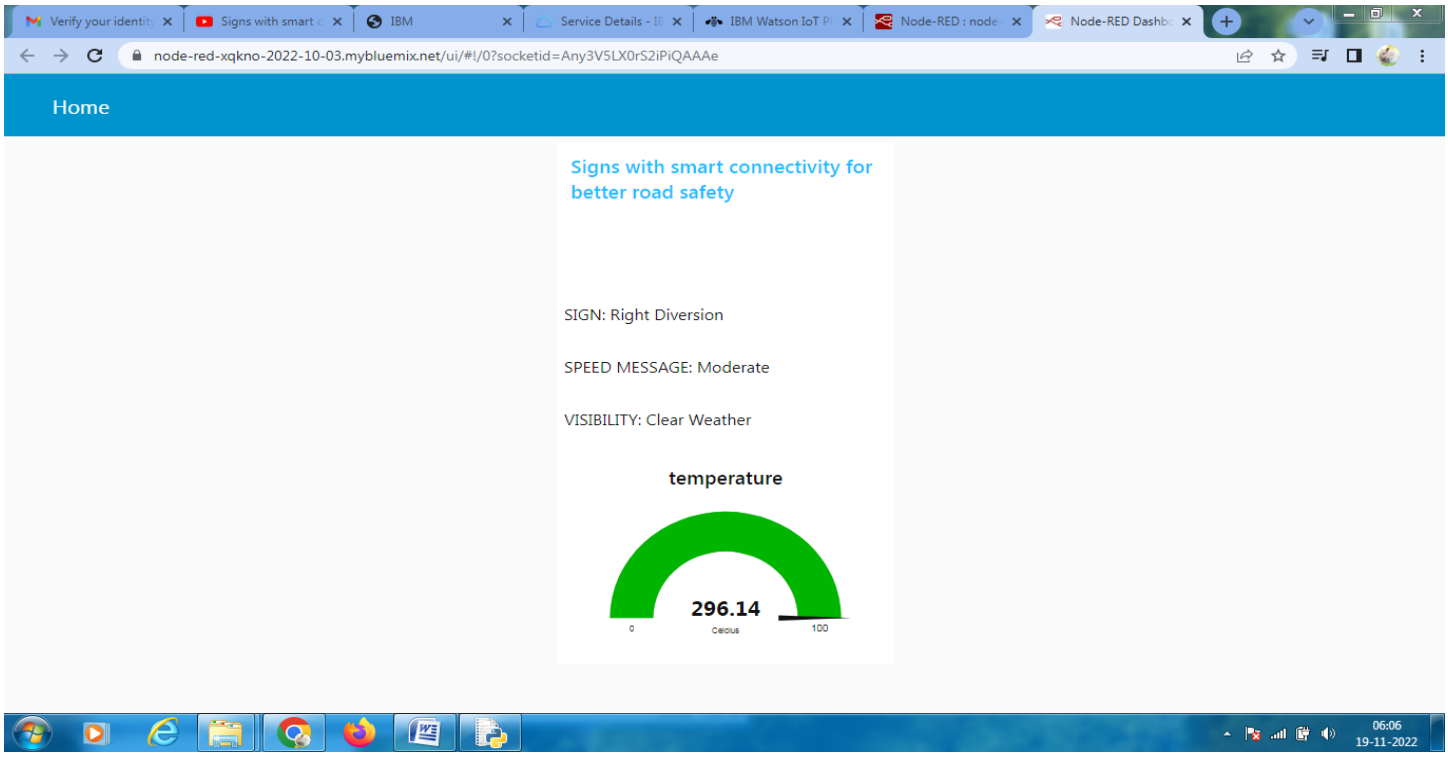
Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end-user. Learning how to write test cases for software requires basic writing skills, attention to detail, and a good understanding of the application under test (AUT).

TestCases:



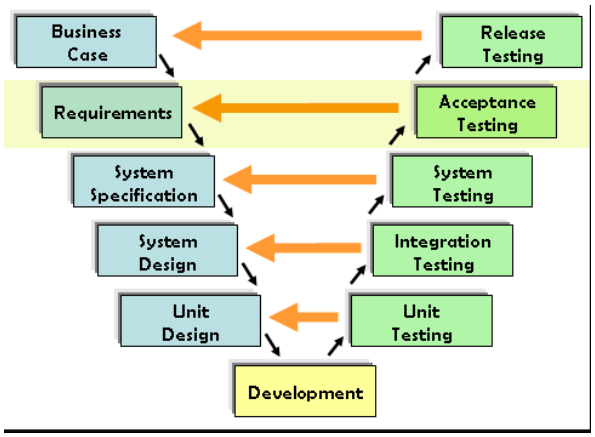
UI : https://74k16v.internetofthings.ibhttps://node-red-xqkno-2022-10-03.mybluemix.net/red/#flow/895a58d22de00831mcloud.com/dashboard/devices/browse





User AcceptanceTesting:

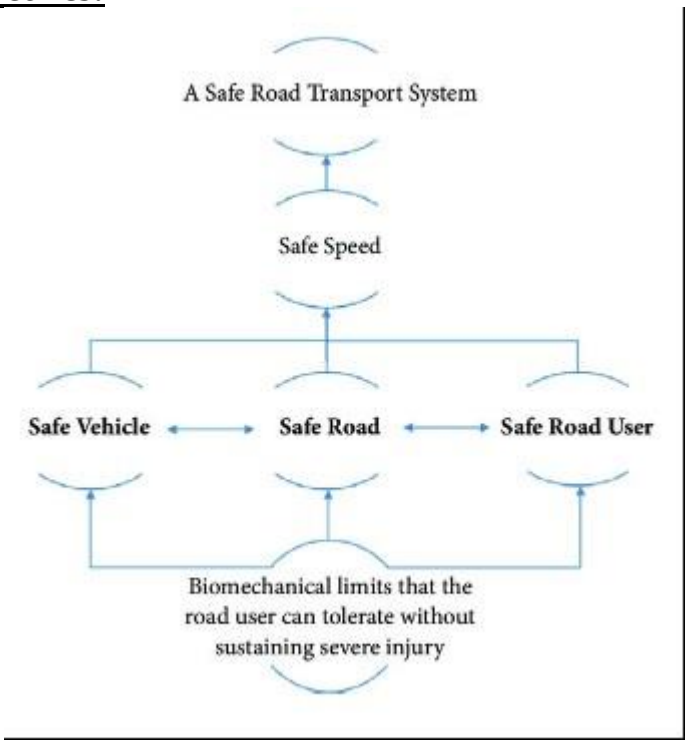
UAT consists, in practice, of people from the target audience using the application. The defects they find are then reported and fixed. This scenario is what most closely resembles "the real world." The process allows users to "get their hands dirty" with the application. They can see if things work as intended**.**



The main purpose of UAT is to validate end-to-end business flow. It does not focus on cosmetic errors, spelling mistakes, or system testing. User Acceptance Testing is carried out in a separate testing environment with a production-like data setup. It is a kind of black box testing where two or more end-users will be involved.

9.RESULTS:

PerformanceMetrics:



10. ADVANTAGES & DISADVANTAGES

Advantages:

Connected vehicles have various benefits such as

- Multimodalsensorsandedgecomputinghelpspeeduptheflowoftrafficwithreal-time processing, reducing congestion andemissions.
- Smart road technology can assist in optimizing trafficflow
- It will manage road conditions, creating a more sustainable environment withincities.
- Improved control and safety can be achieved through IoT-enabled cars. In case of over-speeding, the notification getsdisplayed.
- Ensuring a safe driving experience with real-time assistance, navigation, and even monitoring driving patterns and any emergency. Additionally, along with the state of the traffic, IoT drivers can receive updated information on the state of the roads, i.e., potholes, ice, grade changes, black spots,etc.

DISADVANTAGES:

- Security and privacy. Keeping the data gathered and transmitted by IoT devices safe is challenging, as they evolve and expand in use....
- Technical complexity....
- Connectivity and power dependence....
- Integration....
- Higher costs (time andmoney)

11. CONCLUSION:

The world doesn't change on its own but we humans can change the world to be safe, better, and harmless. Since the road isn't said to be safe let's make it safer with the technologies present and available to us. The Internet of Things is one of the technologies that can lead us to travel on enhanced safe roads. So let's come together to create a better world with no accidents and a smart road for the future generation.

12. FUTURESCOPE:

IoT obtains the majority of its data with the help of connected cars. These incorporate a large number of sensors that establish communication with the cloud, other vehicles, and devices. Thanks to this it provides data and information of great utility for the improvement of road safety. The safe system approach to road safety emphasizes safety by desigh ensuring safe vehicles, road networks, and road users. Evolving towards the future, the road needs to boil with advanced sensors and antenna systems to have peace with the new era.

13. APPENDIX:

Source Code:

```
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = {
   #Configuration
   "identity": {
      "orgId": "3dpjnk",
      "typeId": "Sign_Board",
      "deviceId":"Board_1"
      },
   #API Key
   "auth": {
      "token": "1234567890"
   }
}

#Receiving callbacks from IBM IOT platform
def myCommandCallback(cmd):
   print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
```

```
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()

#OpenWeatherMap Credentials
BASE_URL ="https://api.openweathermap.org/data/2.5/weather?"
CITY = "Nagercoil"
URL = BASE_URL + "q=" + "Chennai" + "&appid=" + "01df65417ab3968e3fc2a38c4aee27bb"

while True:
    response = requests.get(URL)
    if response.status_code ==200:
        data = response.json()
        main = data['main']
        temperature =main['temp']
        humidity = main['humidity']
        pressure = main['pressure']
        report = data['visibility']

        #messge part
        msg=random.randint(0,5)
        if msg==1:
            message="SLOW DOWN, SCHOOL IS NEAR"
        elif msg==3:
            message="SLOW DOWN, HOSPITAL NEARBY"
        elif msg==5:
            message="NEED HELP, POLICE STATION NEARBY"
        else:
            message=""

        #Speed part
        speed=random.randint(0,150)
        if speed>=100:
            speedMsg=" SLOW DOWN, speed Limit Exceeded"
        elif speed>=60 and speed<100:
            speedMsg="Moderate"
        else:
            speedMsg="Slow"

        #Sign part
        sign=random.randint(0,5)
        if sign==1:
            signMsg="Right Diversion"
        elif sign==3:
            signMsg="Left Diversion"
        elif sign==5:
            signmsg="U Turn"
        else:
            signMsg=""

        #Visibility
        if temperature < 50:
            visibility="Fog Ahead, Drive Slow"
```

```
        else:
            visibility="Clear Weather"

    else:
        print("Error in the HTTP request")
    myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg, 'Speed':speedMsg, 'Visibility':visibility}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    #PUBLISHING TO IOT WATSON
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-18272-1659682357

Project demo link :

https://youtu.be/iJvhOudqiV4

THANKYOU