**Project Report Format**

**Signs with Smart Connectivity for better road safety**

|  |  |
| --- | --- |
| **TEAM ID** | **PNT2022TMID45401** |
| **TEAM LEADER** | **K.S.DINESH KUMAR** |
| **TEAM MEMBER** | **U.MOHAMED AASHIQ** |
| **TEAM MEMBER** | **S.POONTHAMILAN** |
| **TEAM MEMBER** | **J.SAMUVEL STEPHENS** |

**1.Introduction:**

**1.1 Project Overview:**

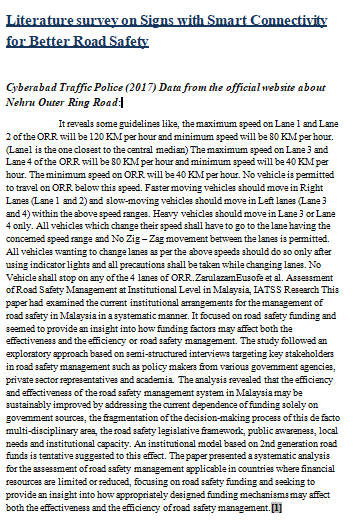
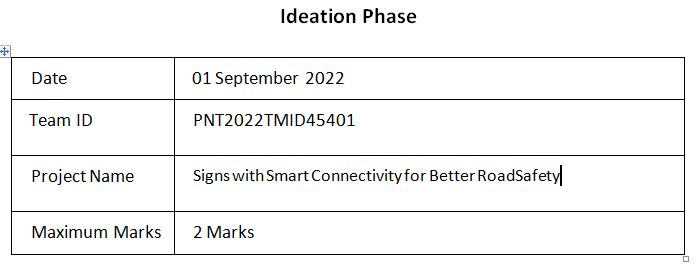
*In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized.This project proposes a system which has digital sign boards 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 of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.*

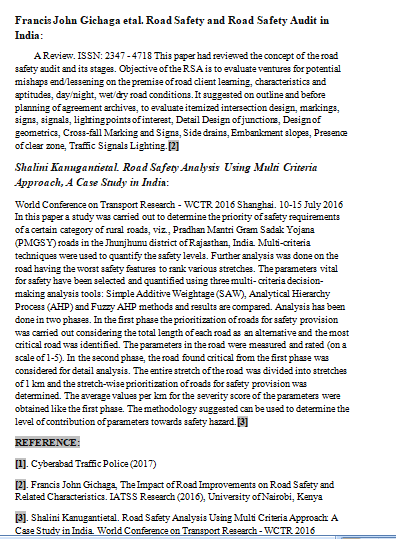
**1.2 Purpose:**

*A Road Safety International task force, comprising leading international experts in road safety and connected mobility, has focused on the relation between interconnected mobility and road safety*

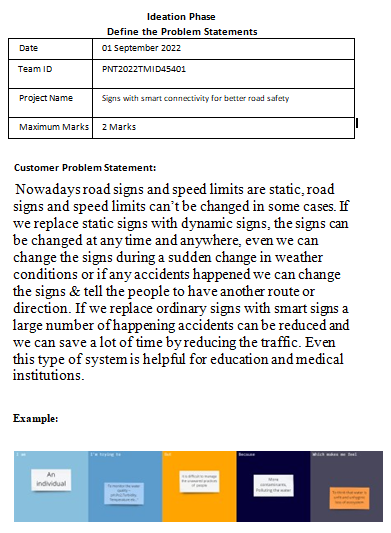
**2.Literature Survey:**

**2.1 Existing Problem & 2.2 Reference:**



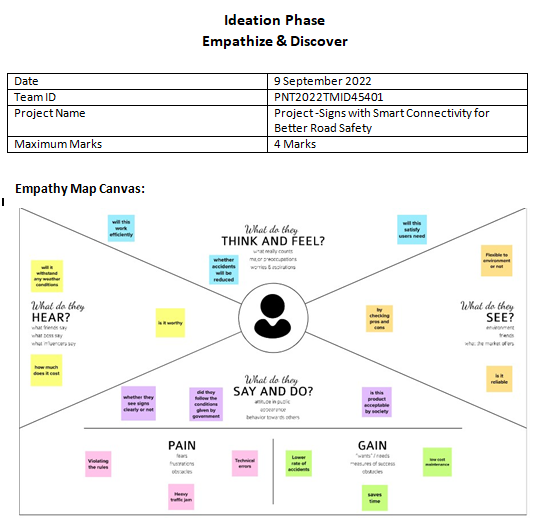


**2.3 Problem Statement Definition:**

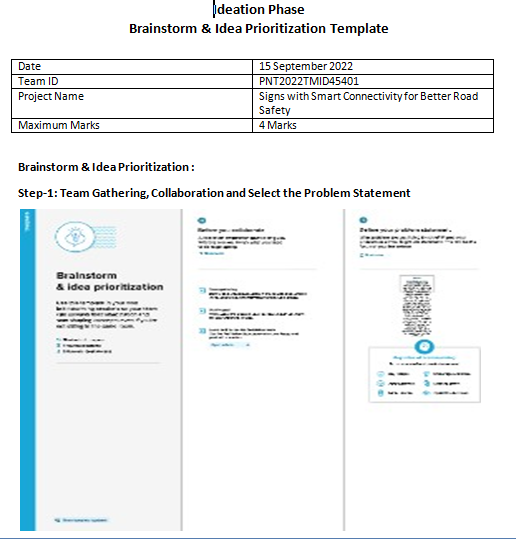


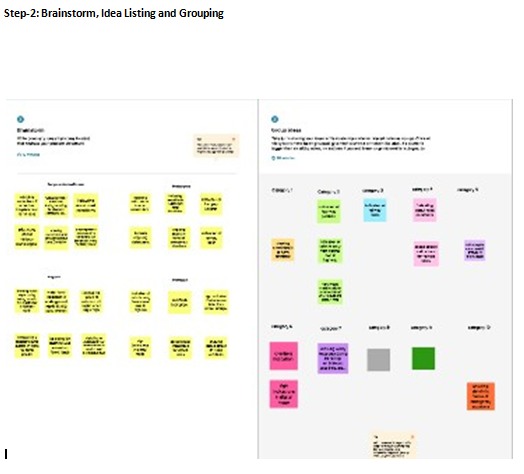
**3.Ideation & Proposed Solution:**

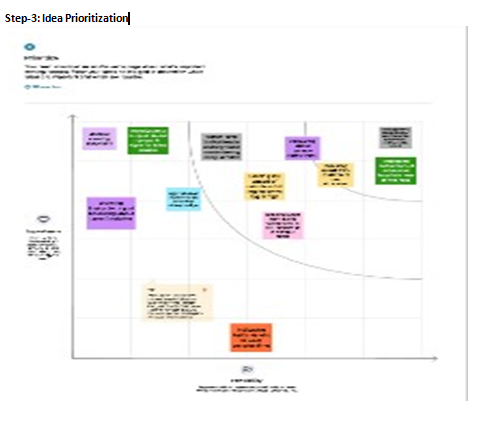
**3.1 Empathy Map Canvas:**



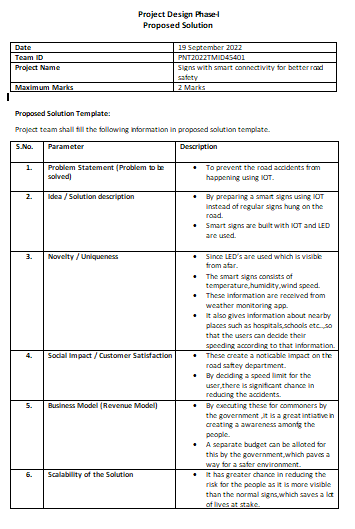
**3.2 Ideation and brainstorming:**





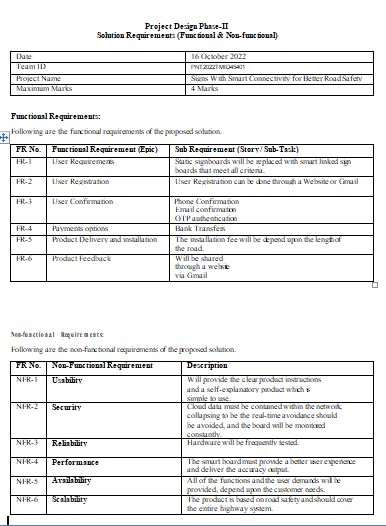


**3.3 Proposed Solution:**



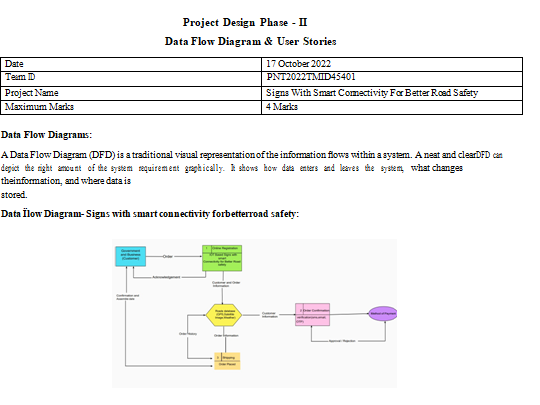
**4.Requirement Analysis:**

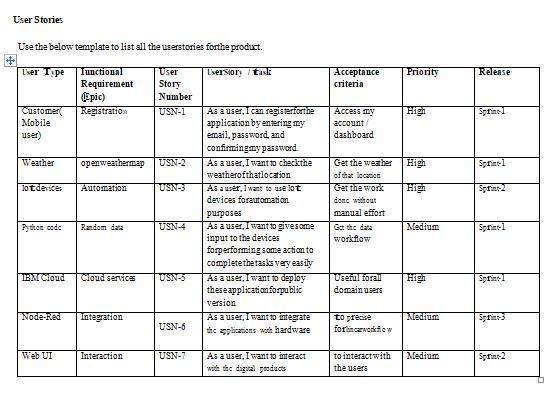
**4.1 Functional Requirements & 4.2 Non Functional Requirements:**

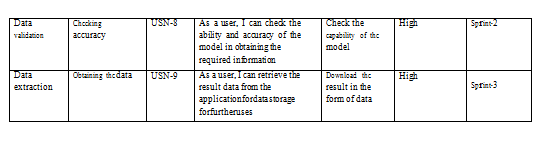


**5.Project Design:**

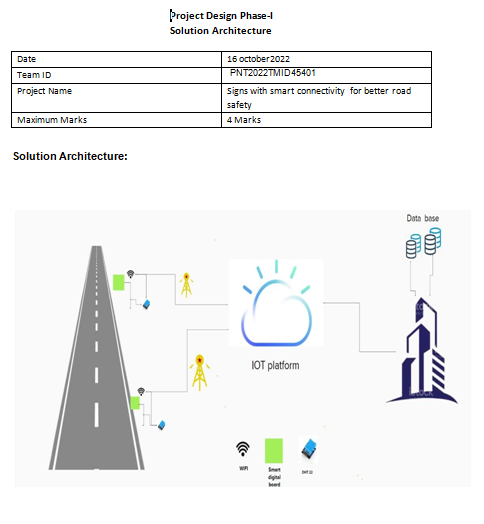
**5.1 Data Flow Diagrams and User Stories:**

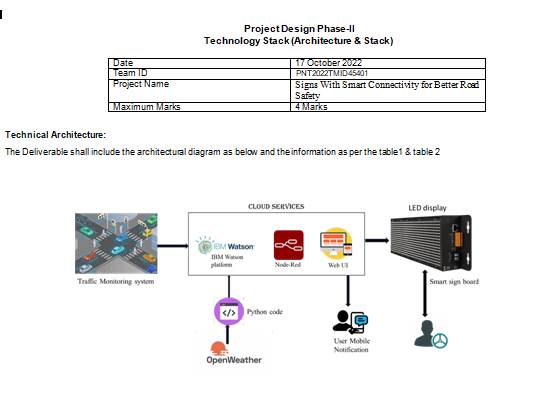


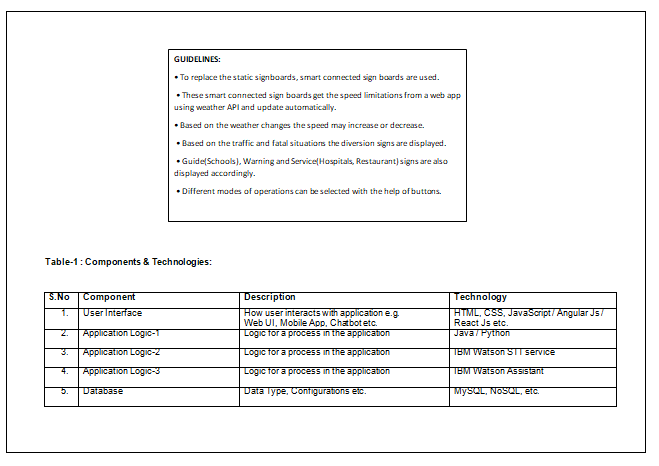


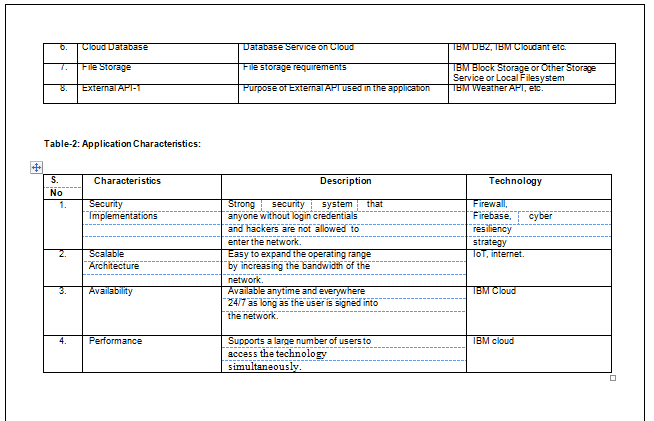


**5.2 Solution and Technical Architecture:**



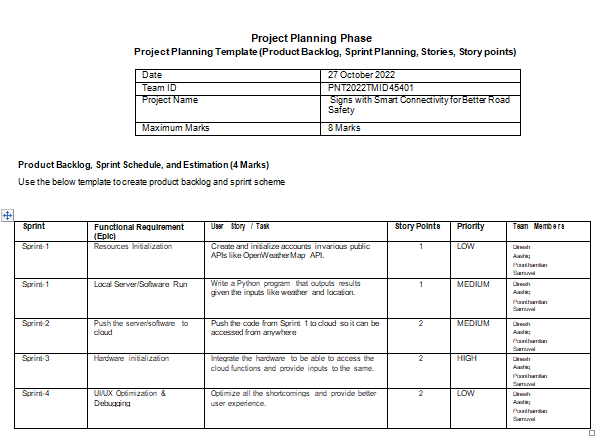


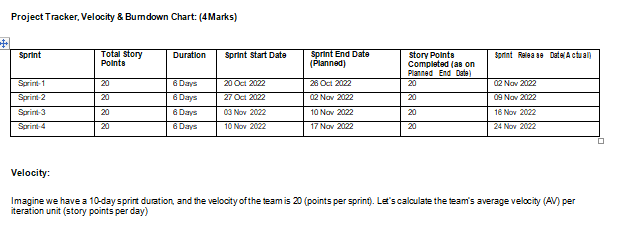


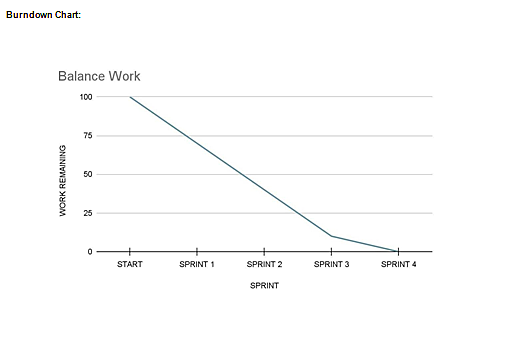


**6.Project Planning & Scheduling:**

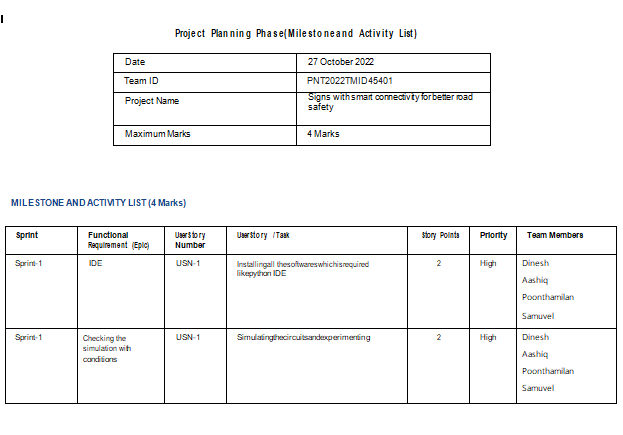
**6.1 Sprint Delivery & Schedule:**

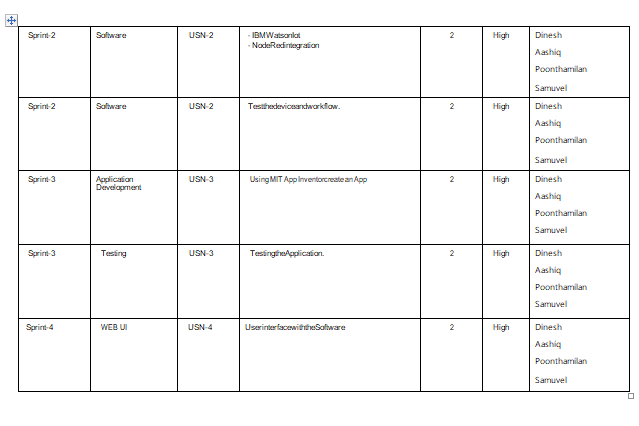




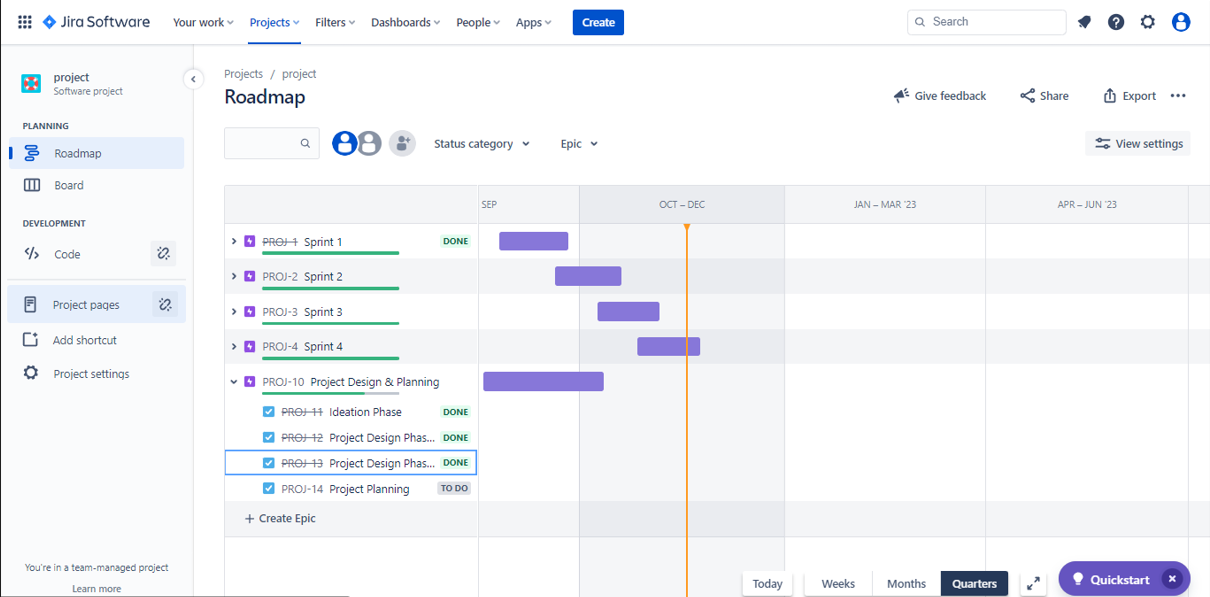


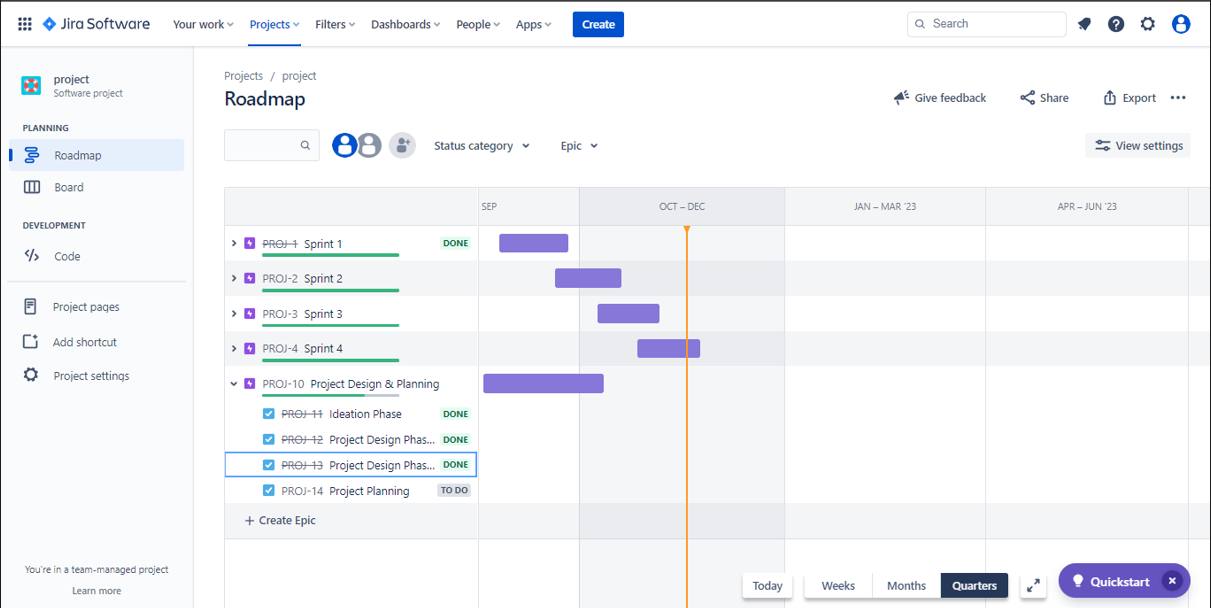
**6.2 Sprint planning & Estimation:**

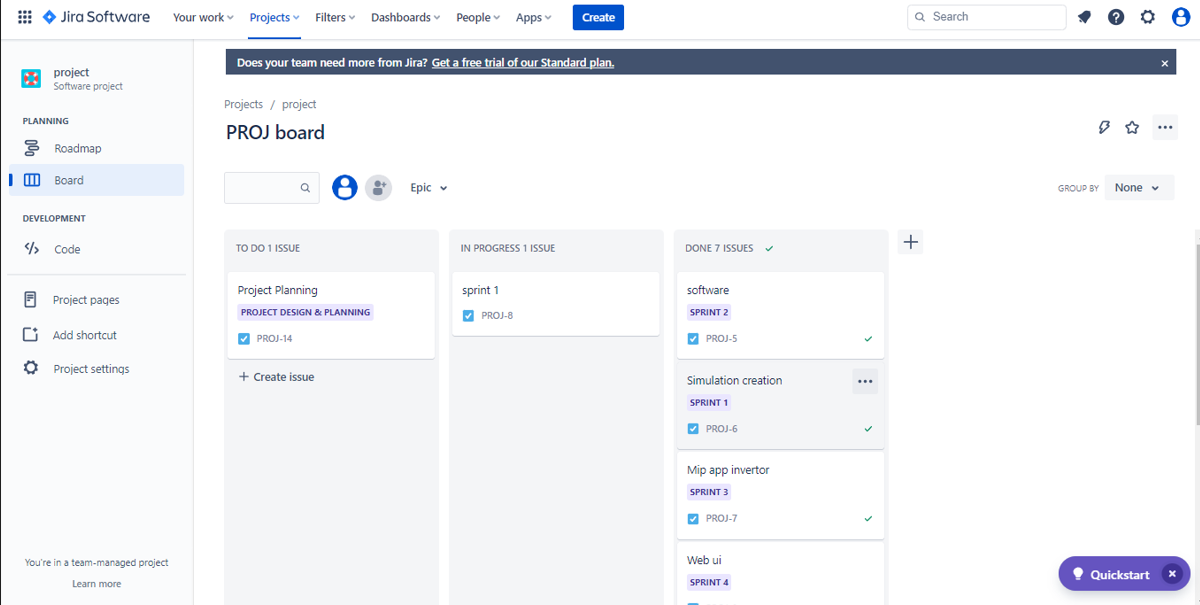




**6.3 Reports from JIRA:**







**7.Coding & Solutioning:**

**7.1 Feature 1:**

*#include<ESP826h> #include<PubSubh> const char\* ssid = "SB-IOT1";*

*const char\**

*password= "sb@iot11"; String command1,command2; #deﬁneORG "bhip5y"*

*#deﬁneDEVICE\_TYPE "Vamsi" #deﬁneDEVICE\_ID "8500"*

*#deﬁne TOKEN "8500913778"*

*String command;*

*char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char topic[] = "iot-2/cmd/home/fmt/String";*

*char authMethod[] = "use-token-auth"; char token[] = TOKEN;*

*char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;*

*/ Serial.println(clientID); #include <Wire.h>*

*#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #deﬁne SSD1306\_LCDHEIGHT 64*

*/ OLED display TWI address #deﬁne OLED\_ADDR 0x3C Adafruit\_SSD1306 display(-1); #if (SSD1306\_LCDHEIGHT != 64)*

*#error("Height incorrect,please ﬁx Adafruit\_SSD1306.h!");*

*#endif*

*void callback(char\* topic,byte\* payload, unsignedint payloadLength); WiFiClient wiﬁClient;*

*PubSubClient client(server, 1883,callback, wiﬁClient); void setup() {*

*display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR); Serial.begin(115200);*

*Serial.println(); pinMode(D1,OUTPUT); wiﬁConnect(); mqttConnect();*

*}*

*void loop() {*

*if (!client.loop()) { mqttConnect();*

*}*

*delay(100);*

*}*

*void wiﬁConnect() {*

*Serial.print("Connecting to "); Serial.print(ssid); WiFi.begin(ssid, password);*

*while (WiFi.status() != WL\_CONNECTED) { delay(500);*

*Serial.print(".");*

*}*

*Serial.print("nWiFi connected, IP address: ");Serial.println(WiFi.localIP());*

*}*

*void mqttConnect() {*

*if (!client.connected()) {*

*Serial.print("Reconnecting MQTT client to "); Serial.println(server); while (!client.connect(clientId, authMethod, token)) { Serial.print(".");*

*delay(500);*

*}*

*initManagedDevice(); Serial.println();*

*}*

*}*

*void initManagedDevice() { if (client.subscribe(topic)) {*

*Serial.println("subscribeto cmd OK");*

*} else {*

*Serial.println("subscribe to cmd FAILED");*

*}*

*}*

*void callback(char\* topic, byte\* payload, unsigned int payloadLength) { Serial.print("callback invokedfor topic: ");Serial.println(topic);*

*for (int i = 0; i < payloadLength; i++) {*

*/ Serial.println((char)payload[i]); command += (char)payload[i];*

*}*

*Serial.println(command); command1=getValue(command,',',0); command2=getValue(command,',',1); if(command1=="1"){ display.clearDisplay();*

*/ display a line of text display.setTextSize(1); display.setTextColor(WHITE); display.setCursor(0,10); display.print(command);*

*/ updatedisplay with all of the above graphicsdisplay.display();*

*}*

*command ="";command1 =""; command2="";*

*}*

*String getValue(String data, char separator, int index)*

*{*

*int found = 0;*

*int strIndex[] = { 0, -1 };*

*int maxIndex = data.length() - 1;*

*for (int i = 0; i <=maxIndex && found<= index; i++) { if (data.charAt(i) == separator || i == maxIndex) {*

*found++;*

*strIndex[0] = strIndex[1] + 1; strIndex[1] = (i == maxIndex) ? i+1 : i;*

*}*

*}*

*return found > index ? data.substring(strIndex[0], strIndex[1]) : "";*

}

**7.2 Feature 2:**

*#include<ESP8266WiFi.h> #include<PubSubClient.h> const char\* ssid = "SB-IOT1";*

*const char\* password= "sb@iot11"; String command1,command2; #deﬁneORG "bhip5y"*

*#deﬁneDEVICE\_TYPE "Vamsi" #deﬁneDEVICE\_ID "8500"*

*#deﬁne TOKEN "8500913778"*

*String command;*

*char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char topic[] = "iot-2/cmd/home/fmt/String";*

*char authMethod[] = "use-token-auth"; char token[] = TOKEN;*

*char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;*

*/ Serial.println(clientID); #include <Wire.h>*

*#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #deﬁne SSD1306\_LCDHEIGHT 64*

*/ OLED display TWI address #deﬁne OLED\_ADDR 0x3C Adafruit\_SSD1306 display(-1); #if (SSD1306\_LCDHEIGHT != 64)*

*#error("Heightincorrect, please ﬁx Adafruit\_SSD1306.h!"); #endif*

*void callback(char\* topic,byte\* payload, unsignedint payloadLength); WiFiClient wiﬁClient;*

*PubSubClient client(server, 1883,callback, wiﬁClient); void setup() {*

*display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR); Serial.begin(115200);*

*Serial.println(); pinMode(D1,OUTPUT); wiﬁConnect(); mqttConnect();*

*}*

*void loop() {*

*if (!client.loop()) { mqttConnect();*

*}*

*delay(100);*

*}*

*void wiﬁConnect() {*

*Serial.print("Connecting to "); Serial.print(ssid); WiFi.begin(ssid, password);*

*while (WiFi.status() != WL\_CONNECTED) { delay(500);*

*Serial.print(".");*

*}*

*Serial.print("nWiFi connected, IP address: ");Serial.println(WiFi.localIP());*

*}*

*void mqttConnect() {*

*if (!client.connected()) {*

*Serial.print("Reconnecting MQTT clientto "); Serial.println(server);*

*while (!client.connect(clientId, authMethod, token)){ Serial.print(".");*

*delay(500);*

*}*

*initManagedDevice(); Serial.println();*

*}*

*}*

*void initManagedDevice() { if (client.subscribe(topic)) {*

*Serial.println("subscribeto cmd OK");*

*} else {*

*Serial.println("subscribe to cmd FAILED");*

*}*

*}*

*void callback(char\* topic, byte\* payload, unsigned int payloadLength) { Serial.print("callback invokedfor topic: ");Serial.println(topic);*

*for (int i = 0; i < payloadLength; i++) {*

*/ Serial.println((char)payload[i]); command += (char)payload[i];*

*}*

*Serial.println(command); command1=getValue(command,',',0); command2=getValue(command,',',1); if(command1=="2"){ display.clearDisplay();*

*/ display a line of text display.setTextSize(1); display.setTextColor(WHITE);*

*display.setCursor(0,10); display.print(command2);*

*/ updatedisplay with all of the above graphicsdisplay.display();*

*}*

*command ="";command1 =""; command2="";*

*}*

*String getValue(String data, char separator, int index)*

*{*

*int found = 0;*

*int strIndex[] = { 0, -1 };*

*int maxIndex = data.length() - 1;*

*for (int i = 0; i <=maxIndex && found<= index; i++) { if (data.charAt(i) == separator || i == maxIndex) {*

*found++;*

*strIndex[0] = strIndex[1] + 1; strIndex[1] = (i == maxIndex) ? i+1 : i;*

*}*

*}*

*return found > index ? data.substring(strIndex[0], strIndex[1]) : "";*

*}*

**7.3 Database Schema:**

*#include <ESP8266WiFi.h> #include <PubSubClient.h> const char\* ssid = "SB-IOT1";*

*const char\* password= "sb@iot11"; String command1,command2; #deﬁneORG "bhip5y"*

*#deﬁneDEVICE\_TYPE "Vamsi" #deﬁneDEVICE\_ID "8500"*

*#deﬁne TOKEN "8500913778"*

*String command;*

*char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char topic[] = "iot-2/cmd/home/fmt/String";*

*char authMethod[] = "use-token-auth"; char token[] = TOKEN;*

*char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;*

*/ Serial.println(clientID); #include <Wire.h>*

*#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #deﬁne SSD1306\_LCDHEIGHT 64*

*/ OLED display TWI address #deﬁne OLED\_ADDR 0x3C Adafruit\_SSD1306 display(-1); #if (SSD1306\_LCDHEIGHT != 64)*

*#error("Height incorrect, please ﬁx Adafruit\_SSD1306.h!");*

*#endif*

*void callback(char\* topic,byte\* payload, unsignedint payloadLength); WiFiClient wiﬁClient;*

*PubSubClient client(server, 1883,callback, wiﬁClient); void setup() {*

*display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR); Serial.begin(115200);*

*Serial.println(); pinMode(D1,OUTPUT); wiﬁConnect(); mqttConnect();*

*}*

*void loop() {*

*if (!client.loop()) { mqttConnect();*

*}*

*delay(100);*

*}*

*void wiﬁConnect() {*

*Serial.print("Connecting to "); Serial.print(ssid); WiFi.begin(ssid, password);*

*while (WiFi.status() != WL\_CONNECTED) { delay(500);*

*Serial.print(".");*

*}*

*Serial.print("nWiFi connected, IP address: ");Serial.println(WiFi.localIP());*

*}*

*void mqttConnect() {*

*if (!client.connected()) {*

*Serial.print("Reconnecting MQTT client to "); Serial.println(server); while (!client.connect(clientId, authMethod, token)) { Serial.print(".");*

*delay(500);*

*}*

*initManagedDevice(); Serial.println();*

*}*

*}*

*void initManagedDevice() { if (client.subscribe(topic)) {*

*Serial.println("subscribeto cmd OK");*

*} else {*

*Serial.println("subscribe to cmd FAILED");*

*}*

*}*

*void callback(char\* topic, byte\* payload, unsigned int payloadLength) { Serial.print("callback invokedfor topic: ");Serial.println(topic);*

*for (int i = 0; i < payloadLength; i++) {*

*/ Serial.println((char)payload[i]); command += (char)payload[i];*

*}*

*Serial.println(command); command1=getValue(command,',',0); command2=getValue(command,',',1); if(command1=="3"){ display.clearDisplay();*

*/ display a line of text display.setTextSize(1); display.setTextColor(WHITE); display.setCursor(0,10); display.print(command2);*

*/ updatedisplay with all of the above graphicsdisplay.display();*

*}*

*command ="";command1 =""; command2="";*

*}*

*String getValue(String data, char separator, int index)*

*{*

*int found = 0;*

*int strIndex[] = { 0, -1 };*

*int maxIndex = data.length() - 1;*

*for (int i = 0; i <=maxIndex && found<= index; i++) { if (data.charAt(i) == separator || i == maxIndex) {*

*found++;*

*strIndex[0] = strIndex[1] + 1; strIndex[1] = (i == maxIndex) ? i+1 : i;*

*}*

*}*

*return found > index ? data.substring(strIndex[0], strIndex[1]) : "";*

**8.Testing:**

**8.1 Test cases:**

*● TEST CASE 1 Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'SLOW DOWN, SCHOOL IS NEAR', 'Sign': '', 'Speed': '', 'Visibility': 'Clear Weather*

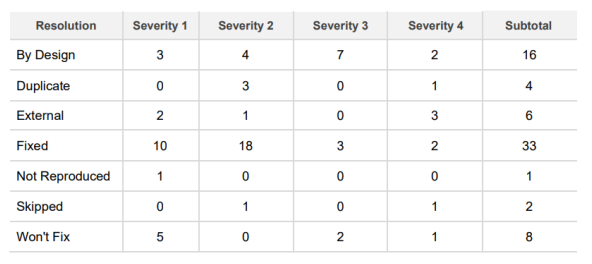
*● TEST CASE 2 Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': '', 'Sign': 'Left Diversion <-', 'Speed': 'SLOW DOWN, Speed Limit Exceeded', 'Visibility': 'Clear Weather*

*● TEST CASE 3 Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'SLOW DOWN, HOSPITAL NEARBY', 'Sign': 'Left Diversion <-', 'Speed': '', 'Visibility': 'Clear Weather*

*● TEST CASE 4 Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'NEED HELP, POLICE STATION NEARBY', 'Sign': 'U Turn', 'Speed': 'Moderate Speed', 'Visibility': 'Clear Weather'.*

**8.2 User Acceptance Testing:**

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



**9.Results:**

**9.1Performance Metrics:**

*The performance of the website varies based on the software chosen for implementation. Built upon NodeJS, a light, and high-performance engine, NodeRED is capable of handling up to 10,000 requests per second. Moreover, since the system is horizontally scalable, an even higher demand of customers can be served.*

**10.Advantages:**

* *Lower battery consumption since processing is done mostly by Node-RED servers in the cloud.*
* *Cheaper and low-requirement microcontrollers can be used since processing requirements are reduced.*
* *Longer lasting systems.*
* *Dynamic Sign updation.*
* *School/Hospital Zone alerts*

**Disadvantages:**

* *The size of the display determines the requirement of the microcontroller.*
* *Dependent on OpenWeatherAPI and hence the speed reduction is the same for a large area in the scale of cities.*

**11.Conclusion:**

*Smart connected Signs for Road Safety, These conclusions and guidelines are addressed to policy makers and private companies that are willing to use innovative solutions to decrease road-related fatalities and injuries amidst populations. Both chapterstake into account the potential users of connected technologies: individual drivers,commercial drivers, pedestrians, cyclists and motorcyclists. The task force decided to study first the potential of connected technologies in high- and middle-income countries. Indeed middle-income countries represent 72% of the World population, 80% of road traffic deaths and 47% of registered motorized vehicles, while high income countries are leaders in development of connected vehicles*

**12.Future Scope:**

1. **Solar powered roadways**

*Photovoltaic cells are embedded within hexagonal panels made of tempered glass, which are used to pave roads. These panels contain LEDs, microprocessors, snow-melting heating devicesand inductive chargingcapability for electricvehicles when driving.Glass is renewable and can be engineered to be stronger than steel, and to allow cars to stop safely even when traveling at high speeds. While this idea has gained widespread support, scalability is a challenge as it remains expensive.*

**2.Smart Roads:**

*Speciallyengineered roadways ﬁtted with smart features, including sensorsthat monitor and report changing road conditions, and WiFi transmitters that provide broadband services to vehicles, homes and businesses. The smart road can also charge electriccars as they drive.*

**3.Glow in the dark roads****:**

*Glowing markers painted onto existing roadway surfaces use a photo-luminescent powder that absorbs and stores daylight. The 500m long strips glow for 8 hours after dark. This technology is still in the testing phase,and the glow is not yet consistent, but it could be more cost-effective than traditional road lighting technologies.*

**4.Interactive lights:**

*Road lights activated by motion sensors to illuminate a particular section of the road as cars approach. The lights dim once the car passes.Suited for roadswith less traﬃc,interactive lights provide night visibility as needed and reduce energy wastage when there are no cars. One design, developed in Holland, uses the wind generated by passing vehiclesto power lights.*

**5.Electric priority lane for chargingelectric vehicles:**

*Embedded cablesgenerate magnetic ﬁeldsthat charge electricvehicles while driving.A receiver coil in the vehicle picks up electromagnetic oscillations from a transmitter coil embedded in the road and converts them to AC, which can then powerthe car. Inductivecharging technology alreadyexists for staticcars, but futurewireless technology could charge batteries while in motion, providing distance range solutions for electric vehicles which travel longerjourneys.*

##### 

##### **6.Weather detection****:**

*Networks of AI-integrated sensors detect weather conditions that impact road safety. Road Weather Information Systems (RWIS) in use today are limited because they only collect data from a small set of weather stations. A larger future network could use automated weather stationsto collect atmospheric and weather data and instantly upload it to the cloud. Dynamic temperature-sensitive paint could be used to highlight invisible roadway conditions like black ice.*

##### **7.Traﬃc detection**

*Data that helps travelers plan their routes. Sensors lining highways monitor traﬃc ﬂow and weight load, warn drivers of traﬃc jams, and automatically alert the authorities about accidents. Fiber-optic cables embedded in the road detect wear and tear, and communication between vehicles and roads can improve traﬃc management. For example, rapid ﬂow technologies use artiﬁcial intelligence (AI) to managetraﬃc lights, which respond to each other and to cars.Traditional systems were pre-programmed to optimize ﬂow around peak journey times, new technologies are able to processand optimize ﬂows in real time.*

**13.Appendix:**

**Source Code:**

*main.py*

*#IBM Watson IOT Platform*

*#pip install wiotp-sdk import wiotp.sdk.device*

*import time*

*import random*

*myConfig = {*

*"identity": {*

*"orgId": "2r52ij",*

*"typeId": "Roadsafety",*

*"deviceId":"1234" },*

*"auth": { "token": "12345678" }*

*}*

*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()*

*while True:*

*temp=random.randint(-20,125)*

*hum=random.randint(0,100)*

*myData={'temperature':temp, 'humidity':hum} client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)*

*print("Published data Successfully: %s", myData) client.commandCallback = myCommandCallback time.sleep(2) client.disconnect()*

**weather.py**

*import requests as reqs*

*def get(myLocation,APIKEY):*

*apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"*

*responseJSON = (reqs.get(apiURL)).json()*

*returnObject =*

*{ "temperature" : responseJSON['main']['temp'] - 273.15,*

*"weather" : [responseJSON['weather'][\_]['main'].lower() for \_ in range(len(responseJSON['weather']))],*

*"visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is 100% and 0km is 0% }*

*if("rain" in responseJSON):*

*returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]*

*return(returnObject)*

**GitHub & Project Link:**

**GitHub: [Github](https://github.com/IBM-EPBL/IBM-Project-33605-1660224485)**

**Project Demo:[Project Demo](https://vimeo.com/manage/videos/772769720)**