V.S.B.ENGINEERING COLLEGE, KARUR

Department of Electronics and Communication Engineering

TITLE : IOT- Safety Gadget For Child Safety Monitoring And Notification

DOMAIN NAME : Internet Of Things

TEAM ID :PNT2022TMI33608

LEADER NAME : Sindhuja J

TEAM MEMBER NAME : Shooriya Prabhaa S
Sharumathi J
Sangeetha R

MENTOR NAME : Nandhini P

ABSTRACT:

Child safety and tracking is a major concern as the greater number of crimes on children are reported nowadays. With this motivation, a smart IoT device for child safety and tracking is developed to help the parents to locate and monitor their children. The system is developed using Link It ONE board programmed in embedded C and interfaced with temperature, heartbeat, touch sensors and also GPS, GSM & digital camera modules. The novelty of the work is that the system automatically alerts the parent/caretaker by sending SMS, when immediate attention is required for the child during emergency. The parameters such as touch, temperature & heartbeat of the child are used for parametric analysis and results are plotted for the same. The above system ensures the safety and tracking of children.

INTRODUCTION:

Internet of Things (IoT) plays a major role in every day-to-day life. The major difference between IoT and embedded system is that a dedicated protocol/software is embedded in the chip in case of embedded system, whereas, IoT devices are smart devices, which are able to take decisions by sensing the environment around the device. The development of sensors technology, availability of internet connected devices; data analysis algorithms make IoT devices to act smart in emergency situations without human interventions. So, IoT devices are applied in different fields such as agriculture, medical, industrial, security and communication applications[1]. IoT systems are useful within a system to do deeper automation, analysis, and integration. IoT contributes to technology by advances in software, hardware and modern tools. It even uses existing and upcoming technology in the fields of sensing, networking and robotics. IoT brings global changes by its advanced elements in the social, economic, and political impact of the users

LITERATURE SURVEY:

The author describes [1] the parent can send a message to the GSM module, according to the message information the GSM module reply back with particular details of the children. The location can be seen on the Google map. When a particular child is facing an emergency situation, device button should be pressed so that the device captures the image along with the user information to the enrolled mobile numbers. The life of the child can be saved within no time.

The author describes [2]a wearable sensor badge is constructed from (hard) electronic components, which can sense perambulatory activities for context awareness. A wearable sensor jacket is used with latest techniques to form (soft) fabric. Stretch sensors are placed to measure upper limb and body movement. Worn as clothing, the sensors give the required information.

The author describes [3]] an analysis of skin resistance and body temperature was made.
Body position is determined by a triple axis accelerometer. After acquiring raw data activity recognition is done and a specialized machine learning algorithm is employed in this process. Real-time data is achieved by sending sensor data to a Cloud Platform. Then the data is analysed using MATLAB. The jacket consists of different sensors for to detect the activity of the body

The author describes[4] there are two modules namely Wi-Fi and audio play back module. The details of the baby can be sent to parents through Wi-Fi module. The audio play back module produces the recorded sound different sensors are accelerometer sensor, cry sensor, temperature sensor gas sensor, flame sensor and PIR sensor. The embedded system consists of microcontroller; accelerometer detects the angular position and movement of the baby.

REFERENCES:

[1] AkashMoodbidri, Hamid Shahnasser (Jan 2017) ‘Child safety wearable device’, International Journal for Research in Applied Science & Engineering Technology, Vol. 6 Issue II, IEEE, pp. 438- 444.

[2] Jonny Farringdon, Andrew J. Moore, Nancy Tilbury, James Church & Pieter Biemond .D (October 1999) ‘Wearable Sensor Badge & Sensor Jacket for Context Awareness’, International symposium on Wearable computers, ISWC 99 proceedings of the 3rd IEEE pp107.

[3] AnandJatti, MadhviKannan, Alisha,RMVijayalakshmi, P ShresthaSinha (May 20-21, 2016), ‘Design and Development of an IoT based wearable device for the Safety and Security of women and girl children’ IEEE International Conference On Recent Trends In Electronics Information Communication Technology, India, pp. 1108-1112

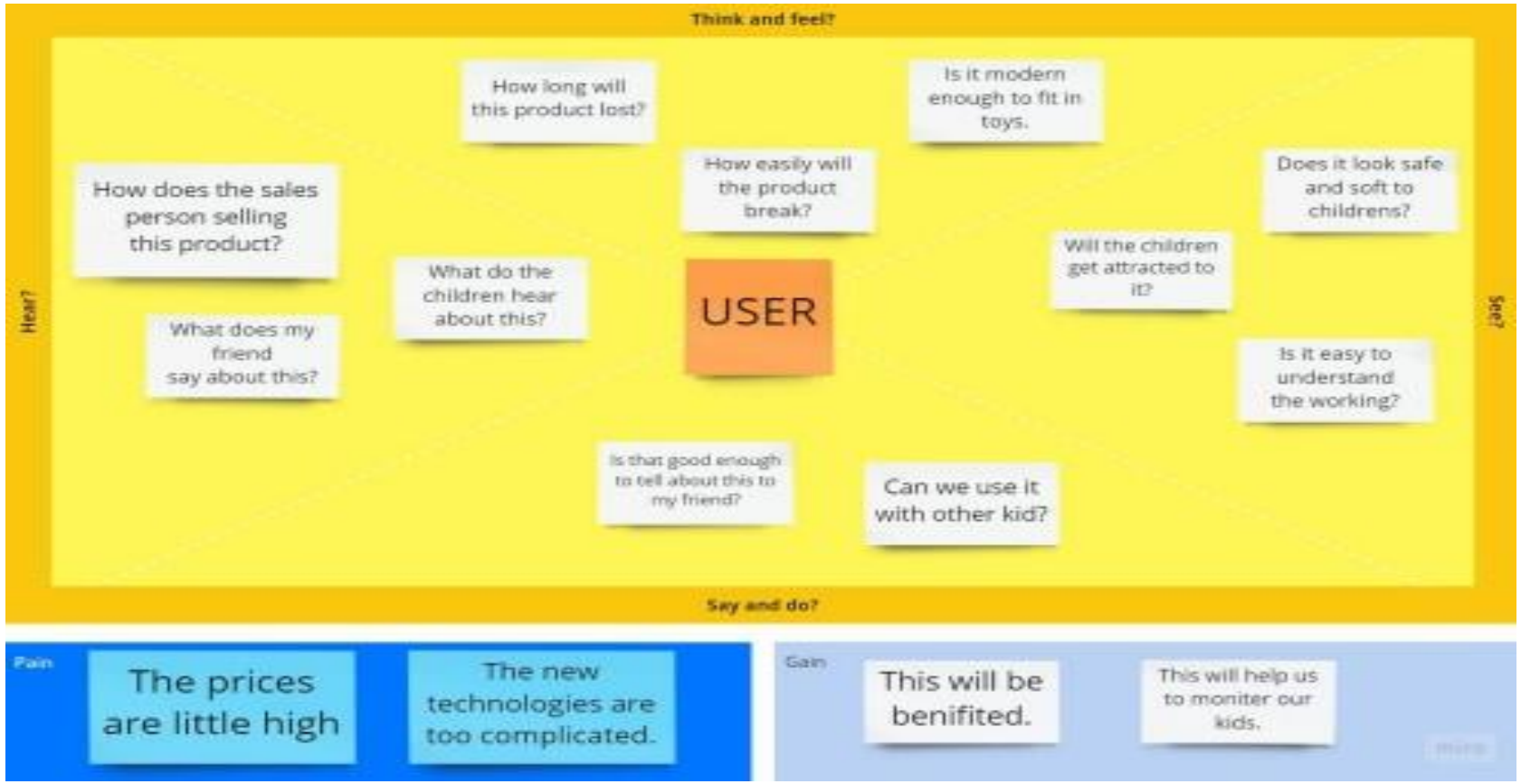
[4] Chitra, jewel jose, sandeep, shirinidhishetty, A. (2018) ‘smart safety jacket for smallbaby’ yenepoyainstittite of technology, moodbidr.

PROBLEM STATEMENT:

THE MAIN OBJECTIVE OF THIS SYSTEM IS TO PROVIDE THE SAFETY TO CHILD WHICH IS LOST IN MAJOR CROWED AREA. NOW A DAY, CHILD ARE NOT SECURED THEY ARE FACING MANY ISSUES REGARDING THEIR SECURITY. THERE ARE NUMBER OF SECURITY SYSTEMS FOR THE CHILD SECURITY PURPOSE. IN ORDER TO OVERCOME SUCH PROBLEMS. THE CHILD SAFETY WEARABLE SYSTEM IS IMPLEMENTED. THIS SYSTEM IS NOT REQUIRED ANY EXPENSIVE TECHNOLOGY AND IT IS USER FRIENDLY FOR BOTH EDUCATED AND UNEDUCATED PEOPLE. THERE ARE MANY WEARABLE DEVICES ARE AVAILABLE IN THE MARKET TO TRACK THE CHILD USING WI-FI AND BLUETOOTH BUT THE WI-FI AND BLUETOOTH ARE THE UNRELIABLE MEDIUM FOR THE COMMUNICATION BETWEEN PARENT AND CHILD. IN THIS SYSTEM WE USE THE TEXT SMS AS A MODE OF COMMUNICATION BETWEEN PARENT AND CHILD THERE IS MINIMUM CHANCES OF FAILING COMMUNICATION AS COMPARED TO WI-FI AND BLUETOOTH. IT ALSO INCLUDES SOS LIGH AND BUZZER TO PROVIDE SECURITY TO THE CHILD IN REAL TIME SITUATIONS AND IT HELPS TO PARENTS TO CHECK THE CONDITION OF CHILD USING ANDROID APPLICATION.

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS:



IDEATION AND BRAINSTORMING:

Sindhuja J

We can use sensors for detecting heartbeat and temperature of the child.

We can use the GPS and GSM totrack the live location.

Sangeetha R

Using mobile mini mobile setupwill help to send the SMS to parents.

We can use sensors to turn onthe mic of the child side.

Sarumathi J

We can use the aurdino formicroprocessing .

We can use the UNO for process

Shooriya Prabhaa S

We can use cloud to store the monitoring data of the children.

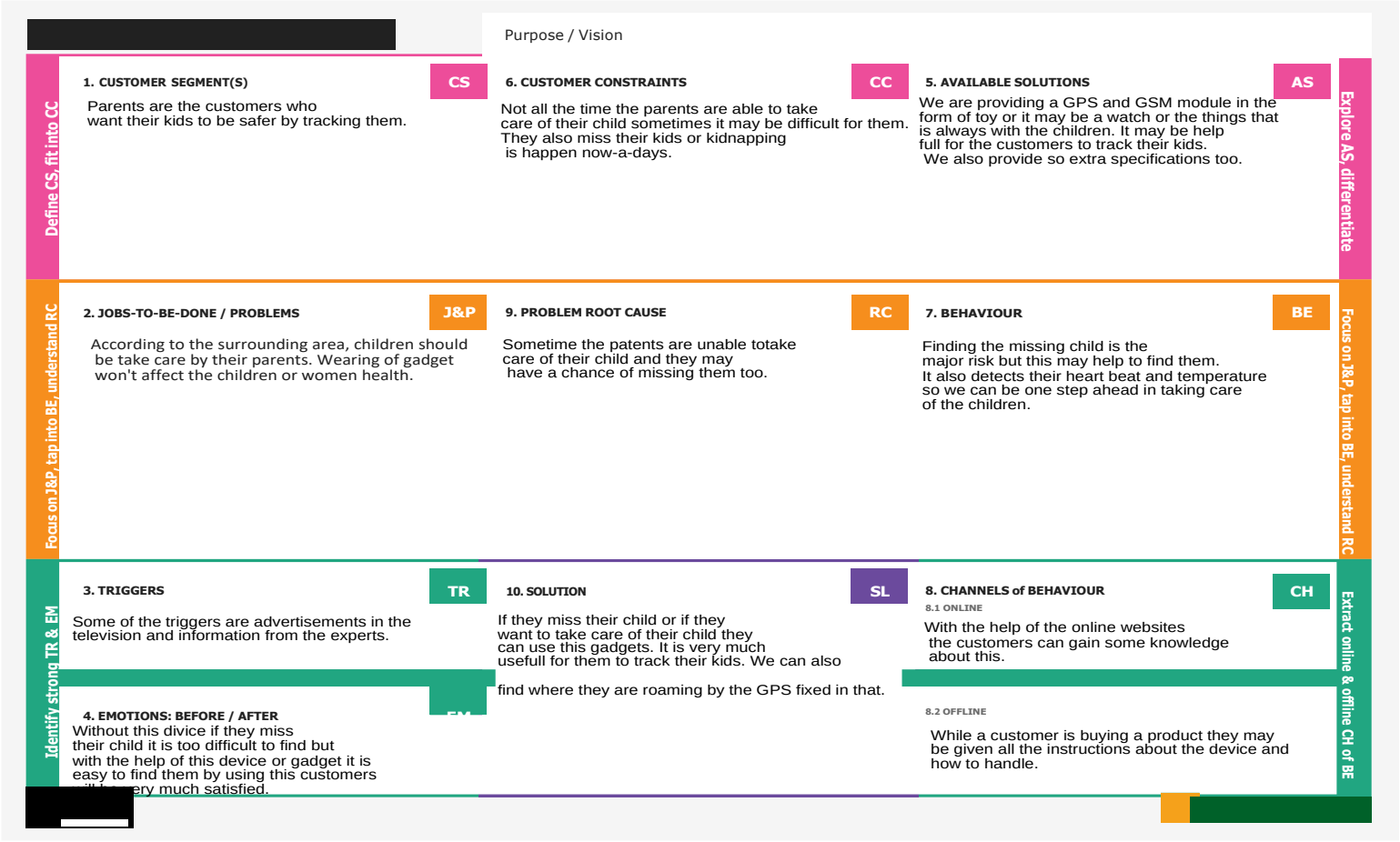
We can use Wi-Fi module to sendthe monitoring data.

PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	As we know in present era everything is based on digital technology. Human being is going to connect each other by using mobile network. This project proposes an SMS based solution to reduced parents insecurity to track children's in real time. Different devices are connected with a single device through. The concerned device is connected to mobile via SMS.
2.	Idea / Solution description	With this motivation, a smart IoT device for child safety and tracking is developed to help the parents to locate and monitor their children.
3.	Novelty / Uniqueness	The system is developed using board programmed in embedded C and interfaced with temperature, heartbeat, touch sensors and also GPS and GMS.
4.	Social Impact / Customer Satisfaction	In this modern world we are not even able to monitor our children this will help the people to monitor even the hearbeat and GPS systems.
5.	Business Model (Revenue Model)	Child and Women play vital roles in our society from their birth to the end of life. In the past few years, crime against child and women has increased to a great extent. They are facing many securities related problems. We can hear the news of women harassments than their achievements.

6.	Scalability of the Solution	It is clearly explained the IoT concept, child safety issues and the need of using child security system. Child safety can be ensured and crime rate will be reduced.

PROBLEM SOLUTION FIT:



REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through SMS Registration through Call Registration through Facebook Registration through Instagram
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via Call

NON-FUNCTIONAL REQUIREMENT:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	*A mobile mini setup will help send the SMS to parents. *We can use sensors to turn on the mic of the child's side.
NFR-2	**Security**	*We can use sensors to detect the child's heartbeat andtemperature. *We can use the GPS and GSM to track the live location.
NFR-3	**Reliability**	*We can use the cloud to store the monitoring data of thechildren. *We can use the wifi modules to send the monitoring data.
NFR-5	**Availability**	*This system is developed using a board programmed inembedded C and interfaced with temperature, and heartbeat. *It is available in online.
NFR-6	**Scalability**	*It is clearly explained the IoT concept, child safety issues, and the need of using a child security System. *Child safety can be ensured and the crime rate will bereduced.

NFR-4	**Performance**	*The novelty of the work is that the system automatically alert the parent /caretaker by sending an SMS when immediate attention is required for the child duringemergency. *The parameters such as touch, temperature, and heartbeatof the child used for parametric analysis..

PROJECT DESIGN:

DATA FLOW DIAGRAM:

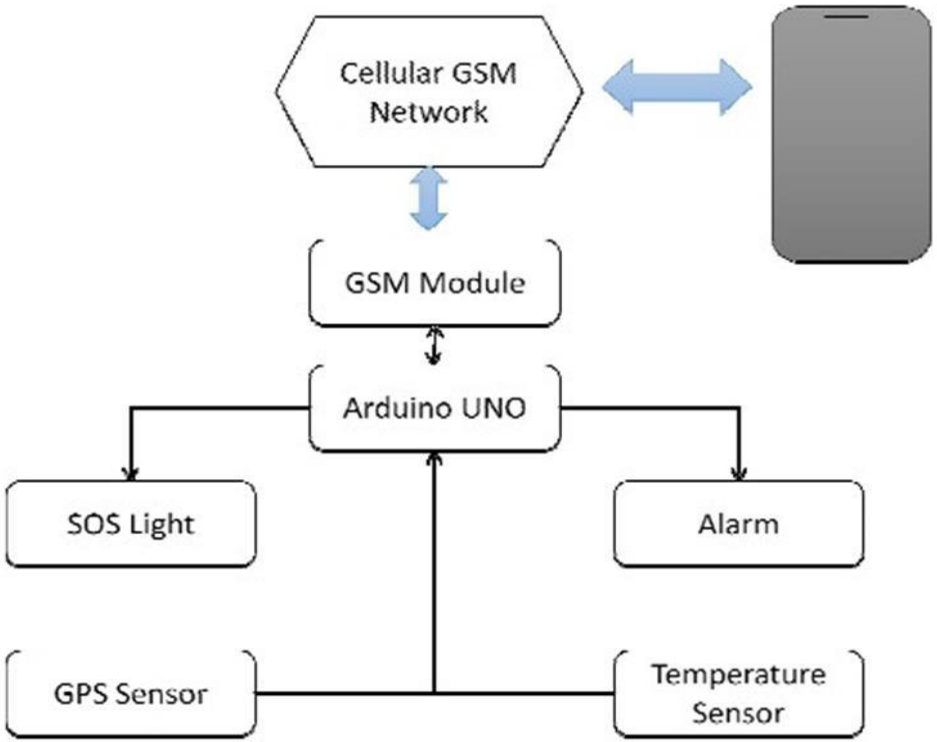


TABLE 1:

S.No	Component	Description	Technology
	User Interface	Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js /React Js etc.
1.	Application Logic-1	Logic for a process in the application	Python
2.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
3.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
4.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
8.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
9.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

TABLE 2:-
APPLICATION AND CHARACTERISTICS:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	*The Internet of Things System (IoT) refers to theset of devices and systems that stay interconnected with real-world sensors and actuators to the Internet	Internet of Things.
2.	Security Implementations	*we can use sensors for detecting heartbeat andtemperature of the child. *We can use the GPS and GSM to track the livelocation.	Sensing technology.
3.	Scalable Architecture	*It is clearly explained the IoT concept ,child safetyissues and the need of using child security System, *Child safety can be ensured and crime rate will bereduced,	Internet of Things
4.	Availability	*This system is developed using board programmed in embedded C and interfaced withtemperature ,heartbeat. *It is available in online.	Microchip technology
5.	Performance	*The novelty of the work is that the system automatically alert the parent/caretaker by sendingSms ,when immediate attention is required for the child during emergency. *The parameter such as touch, temperature andheartbeat of the child are used for parametric analysis,	Infrared temperature sensor.

USER STORIES:

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create the IBM Cloud services which arebeing used in this project.	6	High	Sindhuja.J Sangeetha.R Sarumathi.J Shooriyaprabhaa .S
Sprint-1		US-2	Configure the IBM Cloud services whichare being used in completing this project.	4	Medium	Sindhuja.J Sangeetha.R Sarumathi.J Shooriyaprabhaa .S
Sprint-2		US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM WatsonIoT platform.	5	Medium	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		US-4	In order to connect the IoT device to theIBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	5	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S
Sprint-3		US-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S
Sprint-3		US-2	Create a Node-RED service.	10	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S

Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, heart beat of the child IBMIoT platform	7	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S
Sprint-3		US-2	After developing python code, commandsare received just print the statements which represent the control of the devices.	5	Medium	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa .S
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Sindhuja.J Sangeetha .R Sarumathi. j

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa.S
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	Sindhuja.J Sangeetha.R Sarumathi.J shooriyaprabhaa.S

PROJECT PLANNING AND SCHEDULING

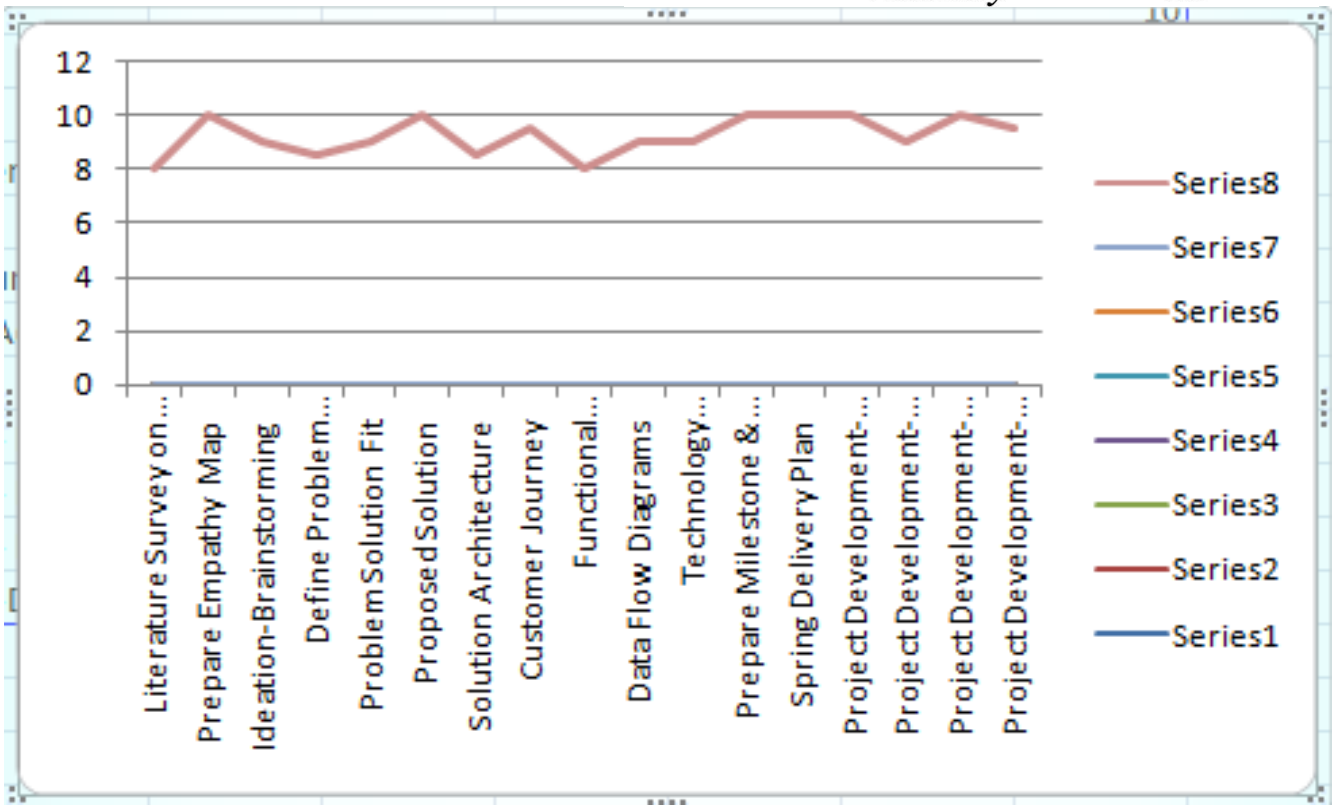
SPRINT PLANNING AND ESTIMATION:

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	25 Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	09 Nov 2022	14 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	16 Nov 2022	21 Nov 2022	20	21 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 (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$



CODING AND SOLUTIONING

FEATURE-1

```
import random
import ibmiotf.application import ibmiotf.device from
time import sleep import sys
#IBM Watson Device Credentials. organization =
"op701j" deviceType = "Lokesh"
deviceId = "Lokesh89" authMethod = "token"
authToken = "1223334444"
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on": print ("sprinkler is ON")
 else :
  print ("sprinkler is OFF") #print(cmd)
```

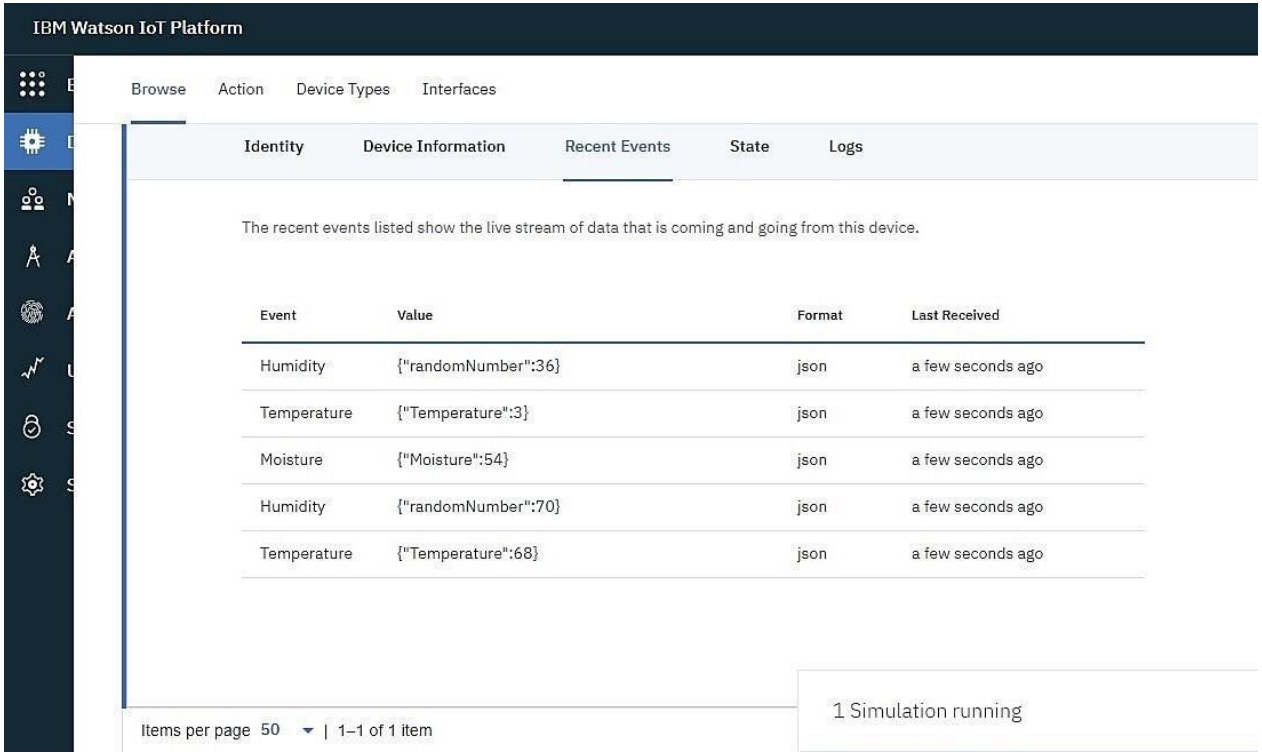
```
try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-
 token": authToken} deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
  print("Caught exception connecting device: %s" % str(e)) sys.exit()
#Connecting to IBM watson.
deviceCli.connect() while True:
#Getting values from sensors.
 temp_sensor = round( random.uniform(0,80),2) PH_sensor =
 round(random.uniform(1,14),3)
 camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
 camera_reading = random.choice(camera)
 flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not
 Detected",] flame_reading = random.choice(flame)
 moist_level = round(random.uniform(0,100),2) water_level =
 round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud. temp_data

 = { 'Temperature' : temp_sensor }
 PH_data = { 'PH Level' : PH_sensor }
 camera_data = { 'Animal attack' : camera_reading} flame_data =
 { 'Flame' : flame_reading } moist_data = { 'Moisture Level' :
 moist_level} water_data = { 'Water Level' : water_level}


# publishing Sensor data to IBM Watson for every 5-10 seconds.
 success = deviceCli.publishEvent("Temperature sensor", "json", temp_data,
 qos=0) sleep(1)
 if success:
  print (" .....publish ok      ")
```

```
print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")

success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)
if success:
 print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")

success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
sleep(1)
if success:
 print ("Published Animal attack %s " % camera_reading, "to IBM Watson")
success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
sleep(1)
if success:
 print ("Published Flame %s " % flame_reading, "to IBM Watson")

success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
sleep(1)
if success:
  print ("Published Moisture Level = %s " % moist_level, "to IBM Watson")
```

```
success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
sleep(1)
if success:
  print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
print ("")
#Automation to control sprinklers by present temperature an to send alert message to IBM Watson.

if (temp_sensor > 35):
  print("sprinkler-1 is ON")
success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinkerlers are turned ON"
%temp_sensor }
, qos=0) sleep(1)
if success:
  print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON" %temp_sensor,"to
IBM Watson") print("")
else:
print("sprinkler-1 is OFF") print("")

#To send alert message if farmer uses the unsafe fertilizer to crops. if

(PH_sensor > 7.5 or PH_sensor < 5.5):
  success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other fertilizer"
  %PH_sensor } ,
qos=0) sleep(1)
if success:
  print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to
IBM Watson") print("")

#To send alert message to farmer that animal attack on crops. if

(camera_reading == "Detected"):
  success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops
detected" }, qos=0) sleep(1)
```

```
if success:
  print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM
Watson") print("")
#To send alert message if flame detected on crop land and turn ON the splinkers to take immediate action.

if (flame_reading == "Detected"):
  print("sprinkler-2 is ON")
success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in
danger,sprinklers turned ON" }, qos=0) sleep(1)
if success:
  print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM Watson")

#To send alert message if Moisture level is LOW and to Turn ON Motor-1 for
irrigation. if (moist_level < 20):
  print("Motor-1 is ON")
success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation started"
%moist_level }, qos=0) sleep(1)
if success:
  print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM
Watson" ) print("")
#To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
```

```
if (water_level > 20):
  print("Motor-2 is ON")
success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON to take water out "
%water_level }, qos=0) sleep(1)
if success:
  print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out "
  %water_level,"to IBM Watson" ) print("")
#command recived by farmer deviceCli.commandCallback =
myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```



FEATURES:

Output: Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor. Power supply: 5V-12V input voltage for most modules (they have a 3.3V regulator),but 5V is ideal in case the regulator has different specs.

BUZZER

Specifications

- RatedVoltage : 6V DC
- Operating Voltage : 4 to 8V DC

- Rated Current*: ≤30mA

- SoundOutput at 10cm* : ≥85dB
- Resonant Frequency : 2300 ±300Hz

- Tone: Continuous A buzzer is a loud noise maker.

Most modern ones are civil defense or air- raid sirens, tornado sirens, or the sirens on emergency service vehiclessuch as ambulances, police cars and fire trucks. There are two general types, pneumatic and electronic.

FEATURE-2:

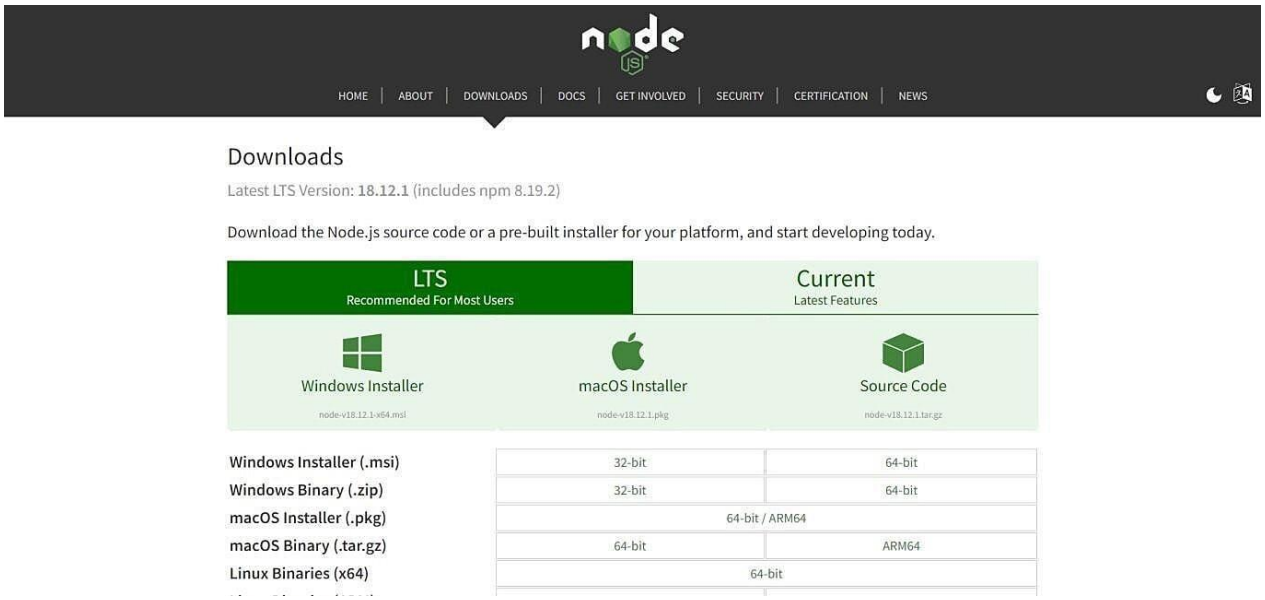
i. Goodsensitivity to Combustible gas in wide range .
ii. Highsensitivity to LPG, Propane and Hydrogen .
iii. Longlife and low cost.
iv. Simpledrive circuit.

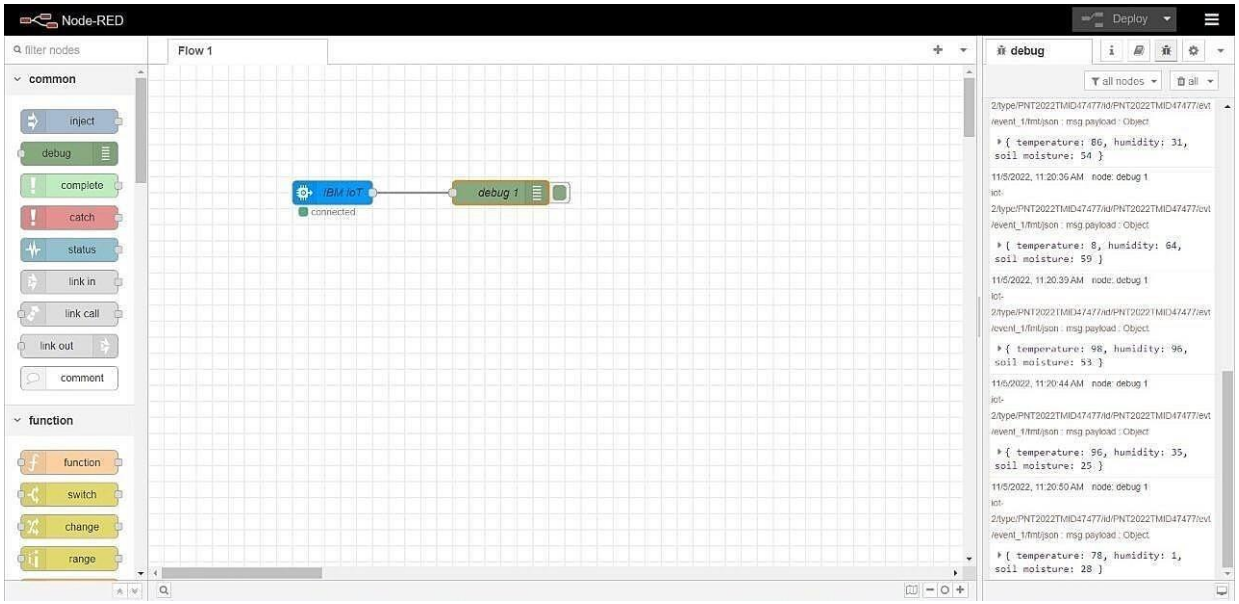
TESTING

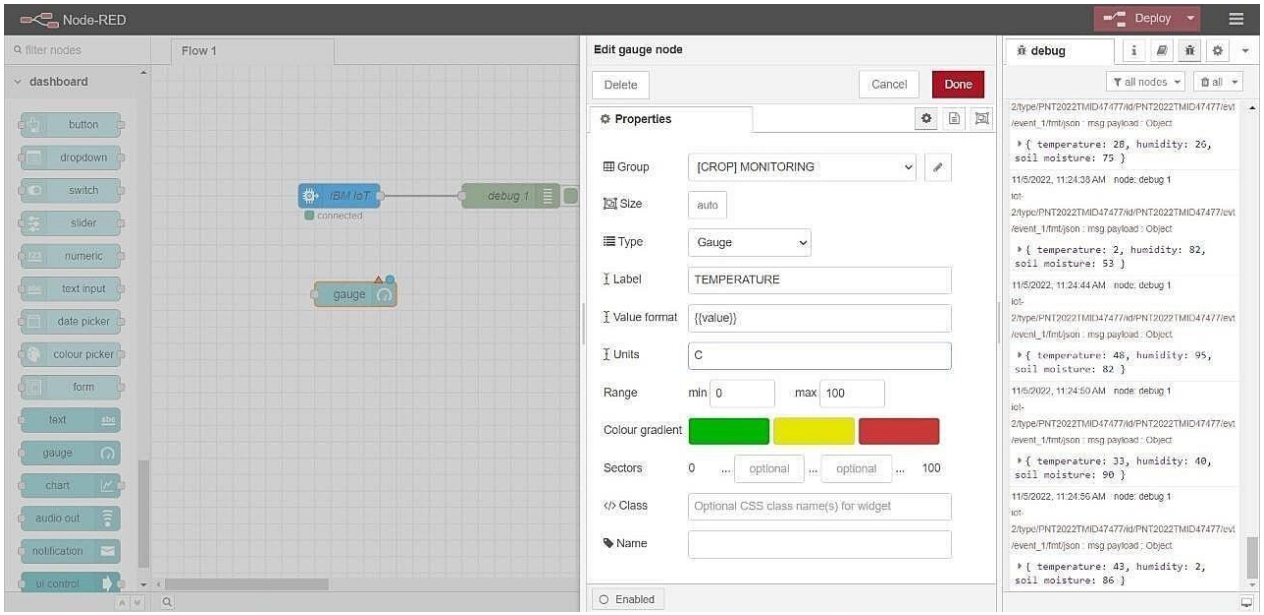
TEST CASES:

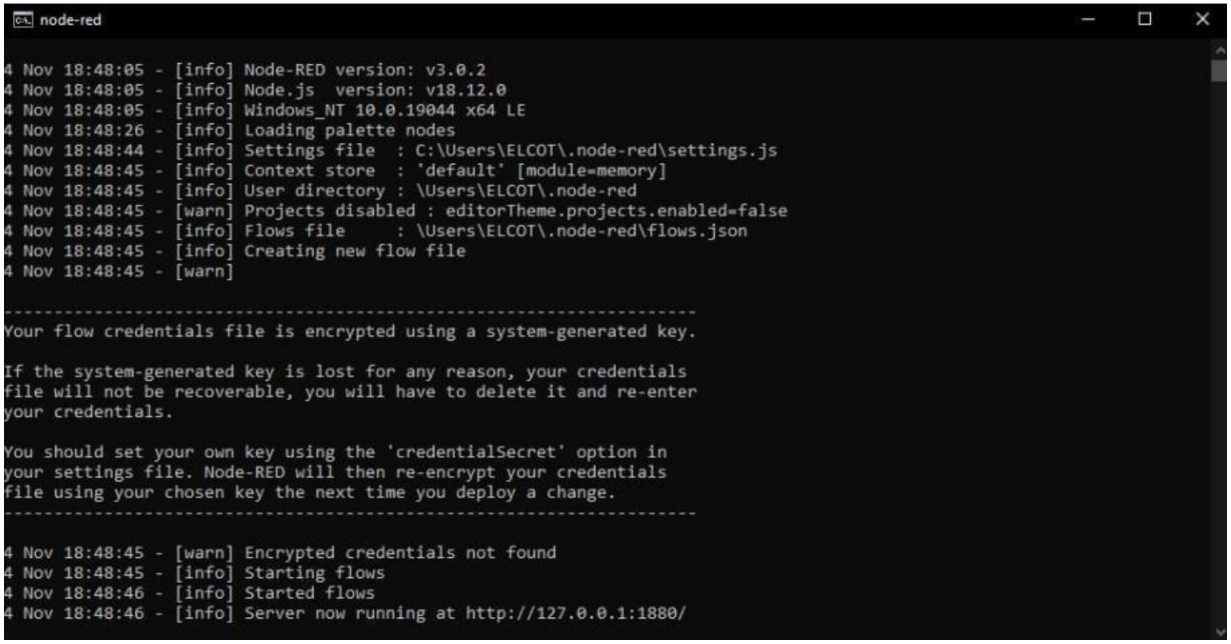
sno	parameter	Values	Screenshot
1	Model summary	-	
2	accuracy	Training accuracy-95% Validation accuracy-72%	
3	Confidence score	Class detected-80% Confidence score-80%	

User Acceptance Testing:




Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
debug 1
debug
all nodes
all
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
{ temperature: 86, humidity: 31, soil moisture: 54 }
11/5/2022, 11:20:36 AM node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
{ temperature: 8, humidity: 64, soil moisture: 59 }
11/5/2022, 11:20:39 AM node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
{ temperature: 98, humidity: 96, soil moisture: 53 }
11/5/2022, 11:20:44 AM node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
{ temperature: 96, humidity: 35, soil moisture: 25 }
11/5/2022, 11:20:50 AM node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
{ temperature: 78, humidity: 1, soil moisture: 28 }


Node-RED
Deploy
filter nodes
Flow 1
dashboard
button
dropdown
switch
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
IBM IoT
connected
debug 1
gauge
Edit gauge node
Delete
Cancel
Done
Properties
Group
[CROP] MONITORING
Size
auto
Type
Gauge
Label
TEMPERATURE
Value format
{{value}}
Units
C
Range
min 0
max 100
Colour gradient
Sectors
0
optional
optional
100
Class
Optional CSS class name(s) for widget
Name
Enabled
debug
all nodes
all
{ temperature: 28, humidity: 26, soil moisture: 75 }
{ temperature: 2, humidity: 82, soil moisture: 53 }
{ temperature: 48, humidity: 95, soil moisture: 82 }
{ temperature: 33, humidity: 40, soil moisture: 90 }
{ temperature: 43, humidity: 2, soil moisture: 86 }


node-red
4 Nov 18:48:05 - [info] Node-RED version: v3.0.2
4 Nov 18:48:05 - [info] Node.js version: v18.12.0
4 Nov 18:48:05 - [info] Windows_NT 10.0.19044 x64 LE
4 Nov 18:48:26 - [info] Loading palette nodes
4 Nov 18:48:44 - [info] Settings file : C:\Users\ELCOT\.node-red\settings.js
4 Nov 18:48:45 - [info] Context store : 'default' [module=memory]
4 Nov 18:48:45 - [info] User directory : \Users\ELCOT\.node-red
4 Nov 18:48:45 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Nov 18:48:45 - [info] Flows file   : \Users\ELCOT\.node-red\flows.json
4 Nov 18:48:45 - [info] Creating new flow file
4 Nov 18:48:45 - [warn]
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
4 Nov 18:48:45 - [warn] Encrypted credentials not found
4 Nov 18:48:45 - [info] Starting flows
4 Nov 18:48:46 - [info] Started flows
4 Nov 18:48:46 - [info] Server now running at http://127.0.0.1:1880/

. RESULTS :

One of the module in our project is temperature sensor which is used to detect the temperature of the child as well as the surrounding temperature. If there occurs any abnormal rise or fall in temperature in the body of the child or in the surrounding it will notify the user as per the coded time delay as shown in the picture. It will show the temperature and humidity values notifies the user based on the predefined value abnormal fall or rise scenarios.

ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1 **Easy Availability& Affordability:**

Gone are the days when buying a GPS enabled Wearable Device for kids was considered a luxury. Today, however, the scenario is different. There are plenty of options readily available. It is easy to buy a smart watch for kids of your choice online. What's more, these magnificent tech gadgets don't burn a big hole in your pockets and make up for an affordable buy. Now a smart watch is just a click away! Besides ,these smart-watches lend a style statement to your fashion conscious kids.

2 **Tracking Made Easy:**

Fueled by IOT, the GPS enabled Wearable Device act as a saviour for parents who are always clouded with worries about their kids. Tracking a child was never this easy. These Wearable Device allow parents to track their children in crowded/public places or when they are out of sight say at school, picnic or an outing. Parents can use these smart-watches to track the location of their lost kids.

3 **Smart watch is Technology in Disguise:**

No matter how tech advanced the smart watches are, they hardly look like one. Most manufacturers have worked hard to mold their tech wonders in a time piece that looks everything but a tech piece! Their childish designs and bright colour combination is perfect to disguise them. This is precisely why most people can hardly spot the difference between

a smart watch and an ordinary watch. Good for kids who use them, as their adorable designs keep these watches safe from the prying eyes.

4 **Watches Over Your Kids:**

GPS tracker watches are a boon for parents as they help in watching over your kids when either they are away or you are away from them. These devices:

1. Tracks kids when they reach school or arrive home from school.

2. Track kids when they are untraceable in a crowded space.

3. Track kids when they are away from home and out of your sight.

5 **Guarantees Peace of Mind to Parents:**

Parents, whether at home or office, are always worried about the safety of their kids. The fear of losing your child to avoidable circumstances is the concern area for all mommies and daddies. On the other hand, a smart watch equipped kid is always traceable and reachable in case of contingencies and emergencies. This in fact, offers great solace for parents, who are relieved at the thought of maintaining an uninterrupted connectivity with their children, anytime, anywhere. Enough to of course, guarantee the much-needed peace of mind.

DISADVANTAGES:

High cost but one it is implemented the expense can be reduced
Battery

CONCLUSION:

The child safety wearable device is capable of acting as a smart IOT device. It provides parents with the real-time location, surrounding temperature, UV radiation index and SOS light along with Distress alarm buzzer for their child's surroundings and the ability to locate their child or alert bystanders in acting to rescue or comfort the child. The smart child safety wearable can be enhanced much more in the future by using highly compact Arduino modules such as the Lilypad Arduino which can be sewed into fabrics. Also a more power efficient model will have to be created which will be capable of holding the battery for a longer time

FUTURE SCOPE:

In our system, we automatically monitor the child in real time using Internet of Things, with the help of GPS, GSM, and Raspberry Pi. This system requires network connectivity, satellite communication, and high-speed data connection when we use web camera and GPS to lively monitor. It is difficult to monitor when there occurs any hindrance to satellite communication or any network issue. There also occurs time delay in video streaming through the server. Hence in the future, these issues can be overcome by using Zigbee concept or accessing the system without internet and using high-speed server transmission.

APPENDIX

SOURCE CODE

```
import time importsys import ibmiotf.application # toinstallpip install ibmiotf
importibmiotf.device


# Provide your IBM Watson Device Credentials organization = "8gyz7t" # replace the ORG
ID deviceType = "weather_monitor" #replace the Device type deviceId = "b827ebd607b5" #
replace Device ID authMethod = "token" authToken = "LWVpQPaVQ166HWN48f" # Replace
the authtoken


def myCommandCallback(cmd):  # function for Callbackif


    cm.data['command'] == 'motoron':


print("MOTOR ON IS RECEIVED")
elif cmd.data['command'] == 'motoroff':print("MOTOR OFF IS RECEIVED") if
cmd.command == "setInterval":


 else:
if 'interval' not in cmd.data:
```

```
printf(“output”)

try:

        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"authmethod": authMethod,
                        "auth-token": authToken}  deviceCli
  = ibmiotf.device.Client(deviceOptions)#
.....


exceptException as e:
        print("Caught exception connecting device: %s" % str(e))sys.exit()


    # Connect and send a datapoint "hello" with value "world" into the cloud as an event oftype "greeting" 10 times
deviceCli.connect()

while True:
        deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud deviceCli.disconnect()
```

SENSOR.PY

```
import time import sysimport ibmiotf.application
importibmiotf.device
import random


# Provide your IBM Watson Device Credentials organization = "8gyz7t" # replace the ORG
ID deviceType = "weather_monitor" #replace the Device type deviceId = "b827ebd607b5" #
replace Device ID authMethod = "token" authToken = "LWVpQPaVQ166HWN48f" # Replace
the authtoken
```

```
def myCommandCallback(cmd):

        print("Command received: %s" % cmd.data['command'])print(cmd)

  try:
                deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
    "auth-method": authMethod, "auth-token": authToken}deviceCli =
    ibmiotf.device.Client(deviceOptions)
                #.....


exceptException as e:
               print("Caught exception connecting device: %s" % str(e))sys.exit()


    # Connect and send a datapoint "hello" with value "world" into the cloud as an event oftype "greeting" 10 times
deviceCli.connect()


while True:
           temp=random.randint(0,1
    00)
    pulse=random.randint(0,100) soil=random.randint(0,100)


           data = { 'temp' : temp, 'pulse': pulse ,'soil':soil}#print data  def
    myOnPublishCallback():
               print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %pulse,"Soil Moisture = %s %%" % soil,"to IBM
    Watson")
```

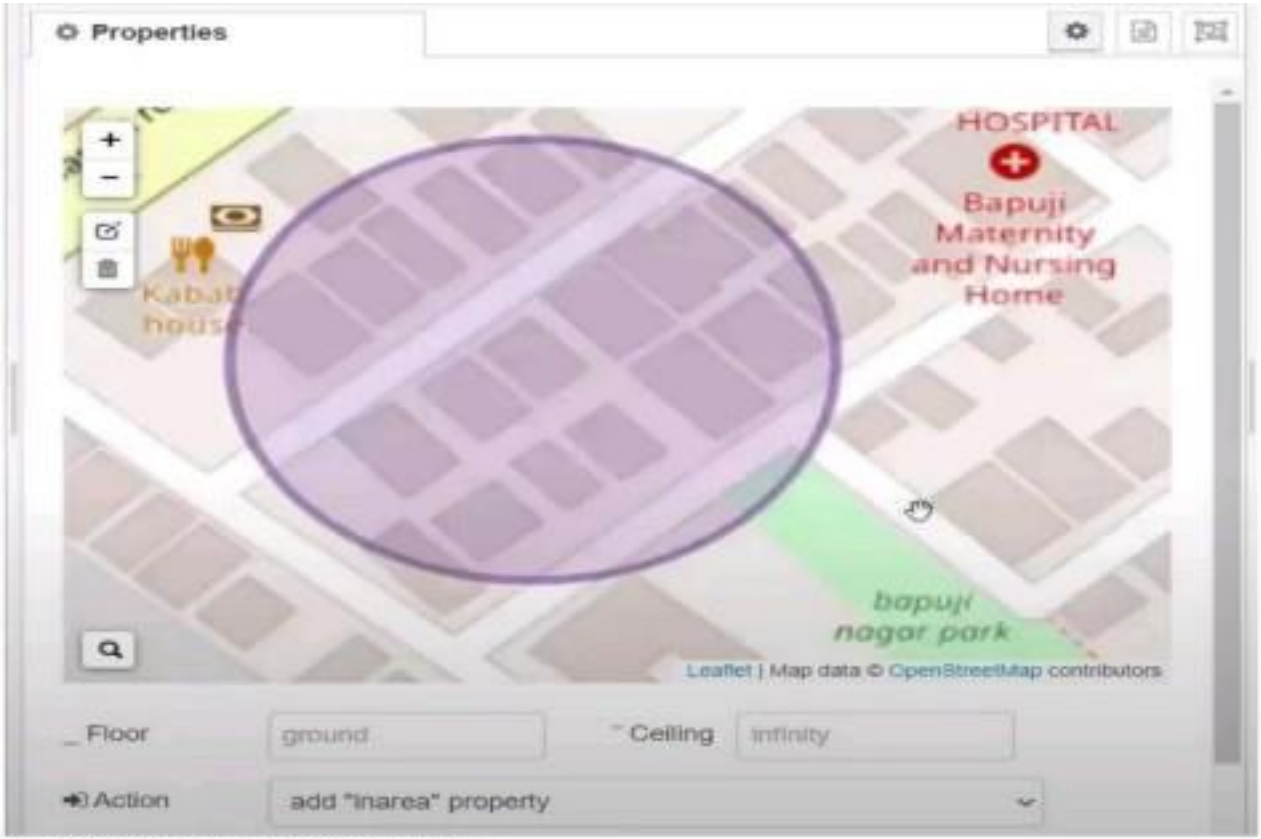
deviceCli.commandCallback = myCommandCallback

Disconnect the device and application from the cloud deviceCli.disconnect()

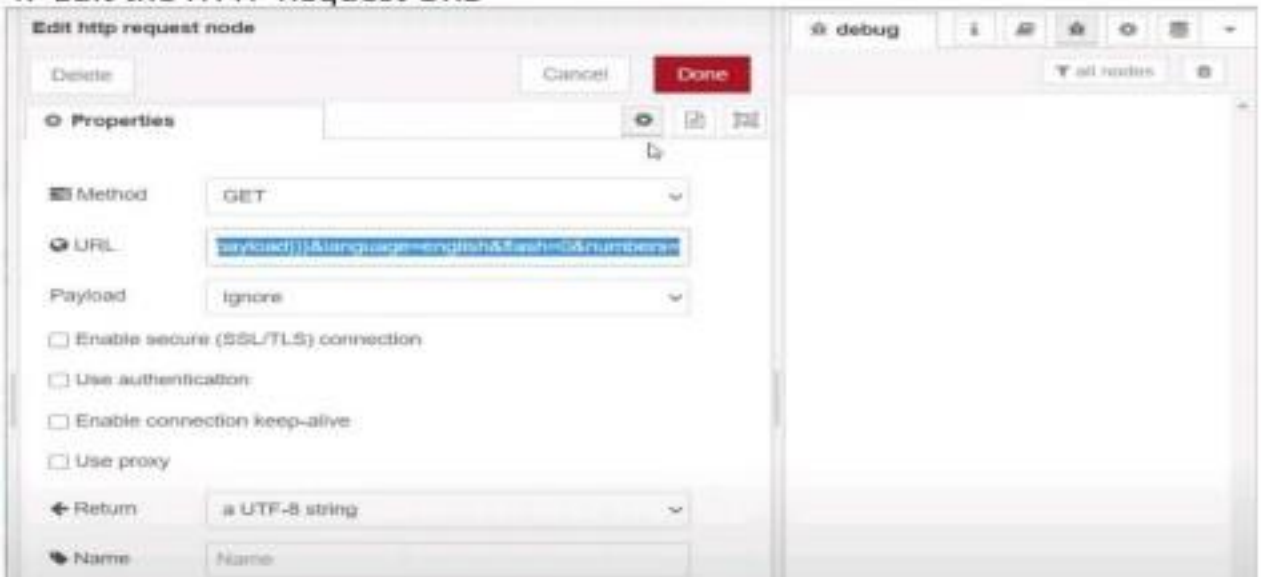
1. Open a Node-RED project

2. Add code to get child location in python

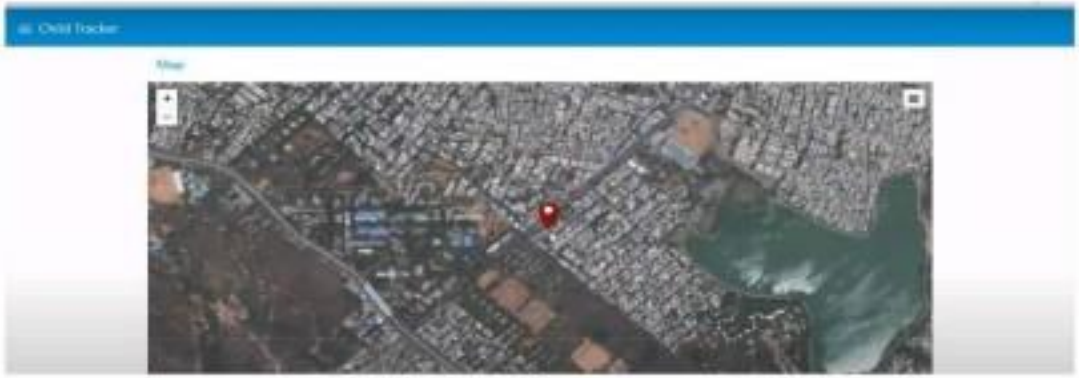
```
import json
import wiotp.sdk.device
import time

myConfig = {
    "identity": {
        "orgId": "hj1fmy",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
        name= "Smartbridge"
        #in area location

        latitude= 17.4225176
        longitude= 78.5458842

        #out area location

        #latitude= 17.4219272
        #longitude= 78.5488783
        myData={'name': name, 'lat':latitude,'lon':longitude}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Data published to IBM IoT platfrom: ",myData)
        time.sleep(5)

client.disconnect()
```



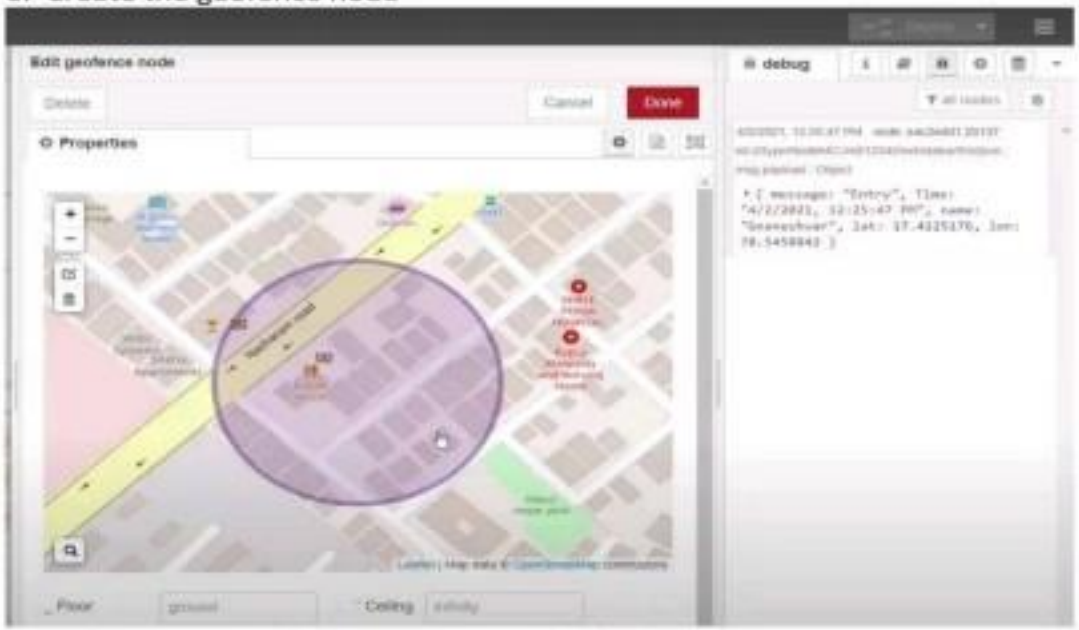
4. Edit the HTTP Request URL



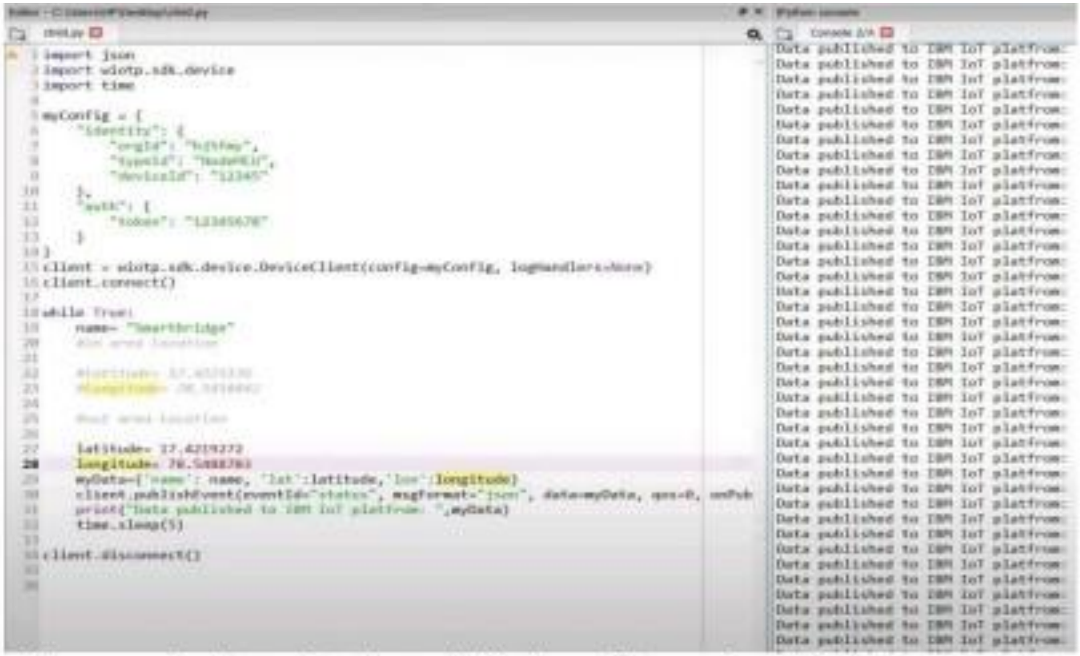
5. Locate the child



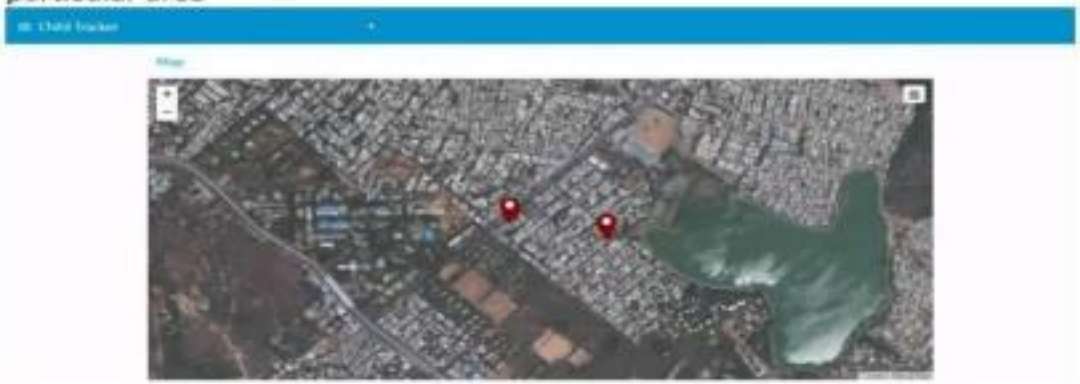
6. Create the geofence node



7. Python script send requests to IBM Cloud



8.After running the script, the web UI shows "Person is not in the particular area"



Conclusion:

Developed the web application using Node-RED Successfully