S

PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

IOT-BASED SAFETY GADGET FOR CHILD SAFETY MONITORING AND NOTIFICATION

A PROJECT REPORT

Submitted by

JEYAPRAKASH P (814719106025)

LOGESSHWARAN R (814719106034)

AZARUDEEN N (814719106011)

JERON JONES J (814719106024)

TEAM ID : PNT2022TMID46033

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SRM TRP ENGINEERONG COLLEGE

(Affiliated to Anna University of Technology)

Irungalur,Mannachanallur(Tk),Trichy(Dt).

NOVEMBER 2022

1. INTRODUCTION

Nowadays attacks on children are increasing at an unprecedented rate and the victims are in dangerous conditions, where they are not allowed to interact the family members. The key knowledge prearranged in this research work is a progressive skill that offers “Smart Child Safety” for the children. Therefore, the awareness of this method is to send an SMS from the children’s wear tool to their parent or guardian. In the prevailing structure, there is no monitoring method for children, which should generate many problems for them, and the no safety mechanism to guard the child against naughtiness. In addition, there is no aware device for the child’s fortification; it must be completed by hand only. Thus, the premeditated method will be highly effective when compared to the other existing techniques in helping the victims. Child chaser helps the parents in unceasingly checking the child's location. They can merely leave their kids in school or parks and create a geofence from place to place the actual location. By continuously examining the child's location warnings will be generated if the child crosses the geofence. Notifications will be sent affording the child's location to their parents or caretakers. The entire location data will be kept in the database. It aims at providing a safe and conducive environment for all children through the prevention and response to child abuse, exploitation, and neglect.

2. LITERATURE SURVEY

2.1. EXISTING WORK

In the real world, children's safety is a huge question mark in everyone’s mind. Parents always expect their children should live in a secure place where they can spend their time and mind without any problem. But, typically half of them are facing so many issues. This issue can be monitored by using IoT components and sensors to check whether people with unaccepted behavior are moving in the child's environment. If children are close to them, then the system has to give an alert message that someone stands with the child. By tracing the children’s locations, the parents can locate where the problem is and how they can help the child with such issues.

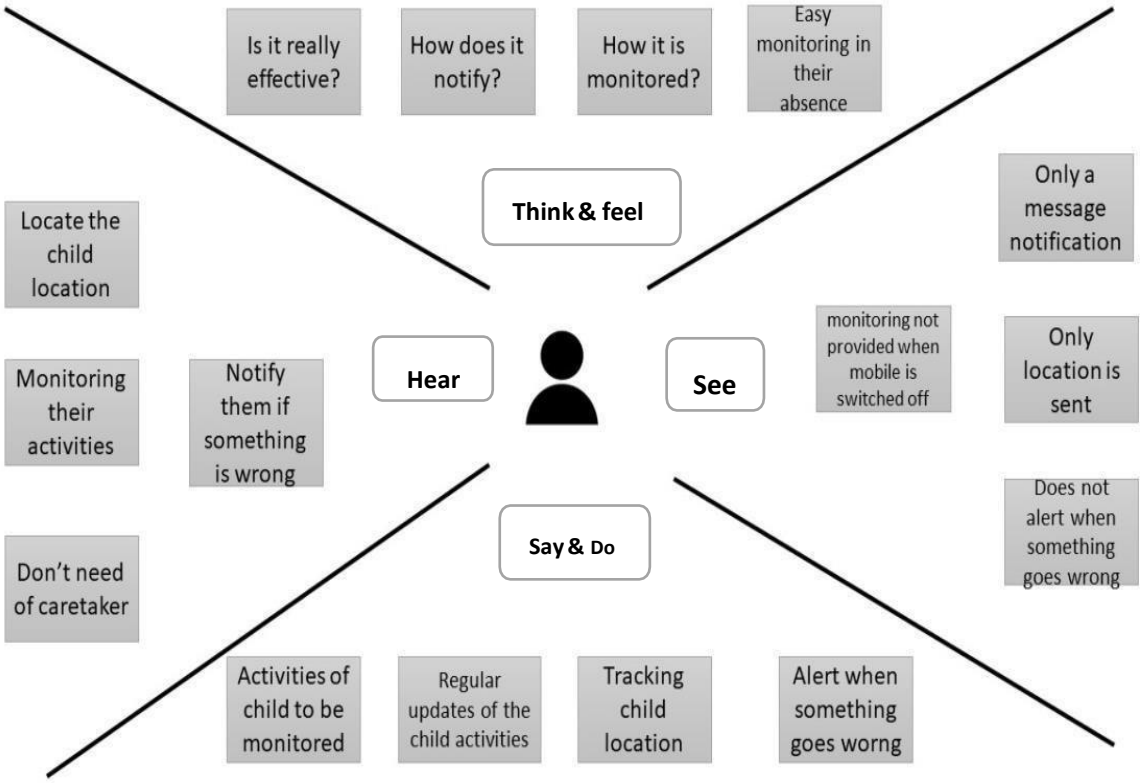
2.2. REFERENCES

[1] Kamat, Mr DK, Ms Pooja S. Ganorkar, and Mrs RA Jain. "Child activity monitoring using sensors." International Journal of Engineering and Techniques 1.3 (2015): 129-133.

[2] Gipsa Alex, Benitta Varghese, Jezna G Jose, AlbyMol Abraham, “A Modern Health Care System Using IoT and Android”, IJCSE, Vol. 8 No.4 Apr 2016.

3.IDEATIONANDPROPOSEDWORK

3.1 EMPATHY MAP



Advantages:

Difficult to track their location if GPS signal is lost

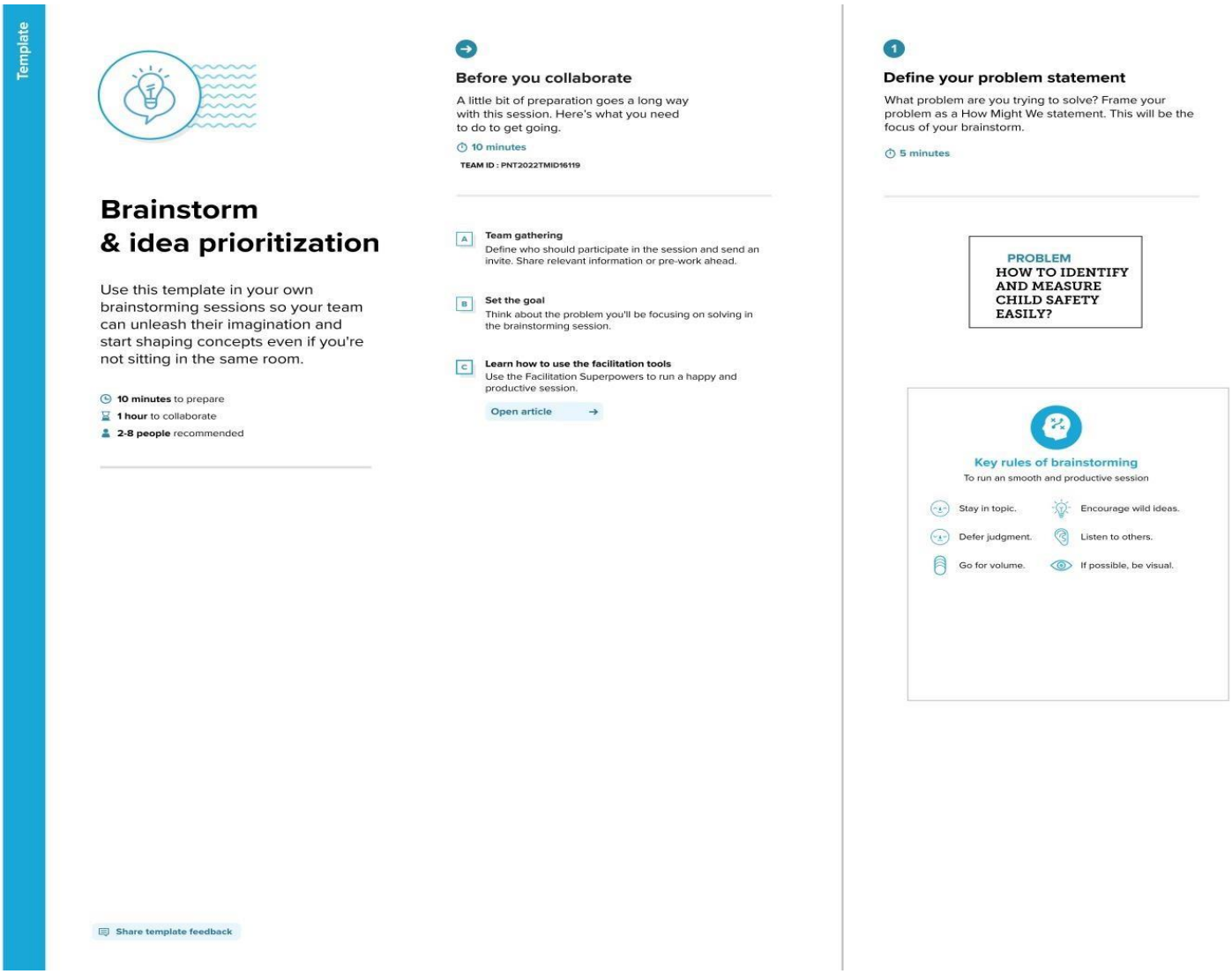
Disadvantages:

Easy to monitor child without a caretaker

3.2. IDEATION AND BRAINSTROMING:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

Logesshwaran .R

Jeyaprakash.P

Jeron Jones.J

Azarudeen.N

Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

3.3. PROPOSED SOLUTION:

Problem Statement:

Currently, parents concern more about serious cases such as missing children, snatching and abuse. They cannot sit with their children or 24*7 hours to protect their children and monitor the children's activities.

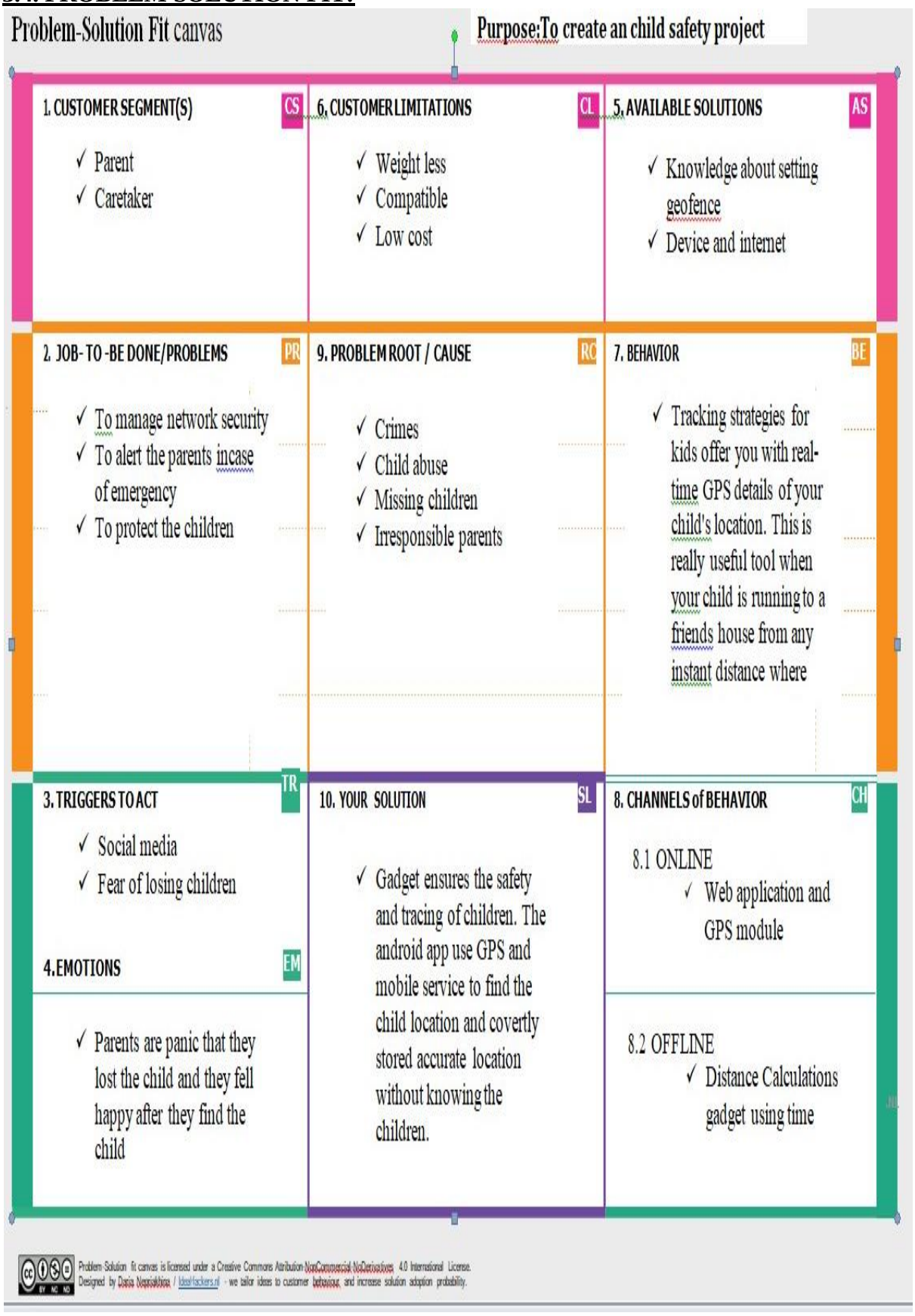
Proposed Solution:

Create a Child tracker which helps the parents with continuously monitoring the child's location. The notification will be sent according to the child's location to their parents or caretakers. The entire location data will be stored in the database. The system automatically alerts the parent/caretaker by sending notification, when immediate attention is required for the child during emergency it can make children parents more assure about their kid's security, we have a feature in our device called Geo-Fence. Geo-Fencing feature allows you to mark a particular area as safe-zone. Whenever your child crosses that specific area, you will get an instant notification on your phone.

Advantages:

- ✓ Easy to use
- ✓ Compatible
- ✓ Weight less

3.4. PROBLEM SOLUTION FIT:



4.REQUIREMENT ANALYSIS:

4.1.FUNCTIONAL REQUIREMENTS

Functional Requirements are:

1. User registration and confirmation
2. App Installation
3. Settings geofence
4. Detecting Child location
5. User interface
6. Database
7. Server
8. GPS Tracking
9. API
10. React JS
11. GPS modules
12. Battery life
13. Location history

USER REGISTRATION AND CONFIRMATION:

Registration through Gmail and Phone number.Confirmation via Email and OTP.

APP INSTALLATION:

Installation through link and playstore.

SETTINGS GEOFENCE:

Setting by user to find child location.

DETECTING CHILD LOCATION:

Detecting location via app and sms.

USER INTERFACE:

User login form and admin login form.

DATABASE:

Stored in cloud for seamless connectivity. Parents and kids link with the distance and the location values obtained from the mobile devices are stored here. The values include parent id,kid id,distance,longitude,latitude etc.

SERVER:

It connects the database and the front end application. The backend server has been implemented to run as a service and is deployed in an IBM cloud instance. The backend server has been implemented to run as a service and is deployed in an IBM cloud instance.

GPS TRACKING:

The system is implemented with a GPS module, which acquires the location information of the user and stores it to the database.

API:

The value collected is sent to the database using an API.

REACT JS:

We are using react js as front end for our project. Node JS for the back end we are using node js.

GPS MODULES:

It receives data directly from satellites.

BATTERY LIFE:

If the child or parent forgets to charge the device for a whole day then also the device will work. That's why we aim to make this device last the whole day with one charge.
It should be long-lasting.

LOCATION HISTORY:

The location history will help to track the child's activity so that the aren't will be updated. Location history will be there for 30 days. For example if the child gets missing with the help of location history the aren't can track down their child's activity and also can find their child.

1) Live Location Tracking:

GPS installed in the device is used to trace the contemporary location and we can keep tabs on it through the android app and SMS requests sent from the safety gadget to the parent’s phone. The child's precise locations are found by parents through the Wearable gadget which in turn employs Global Positioning System to track real-time locations. The software along with relinquishing it allows you to trace down your wards when they’re within Bluetooth limit, it also works when your kids go farther afield. Its adroitness

as a tracker is exceptional if you live in densely colonised neighbourhoods like cities.

2) Panic Alert Systems:

The panic alert mechanism on the device is set off during emergencies; the system software involuntarily alerts the parent/guardian by redirecting a text message where expeditious scrutinization is essential for the child during a catastrophe. The alert is also refurbished to the cloud for the motive of app monitoring.

3) Ceaseless Surveillance:

The gadget ensures utmost security and ensures live tracking for their kids. The device instills child safety through smartphones that can track their children's location and give the precise coordinates of the child's location in real-time anywhere. By monitoring the activities the security state of the child is examined.

4) Cloud Database:

The safety device is equipped with GSM and GPS modules for sending and receiving calls, and SMS between the gadget and the parental phones. The system also consists of a Wi-Fi/cellular data module used to implement IoT and 11send all the monitored parameters to the cloud for android app monitoring on the parental phones. The panic alert system is used during panic situations alerts are sent to the parental phone, seeking help also the alert parameters are updated to the cloud. The history of the location can be stored in the cloud. The wearable devices should feature the child's exact locations and be updated continuously without being interpreted in the cloud database.

1) Security Implementations:

To activate the alarm and facilitate video recording whenever the emergency button is pressed. We can use the cloud to accumulate the surveillance data of the children. The wifi modules are of assistance in sending the monitoring particulars, the user will be notified with an update if any errors are found, for the efficient functioning of the device.

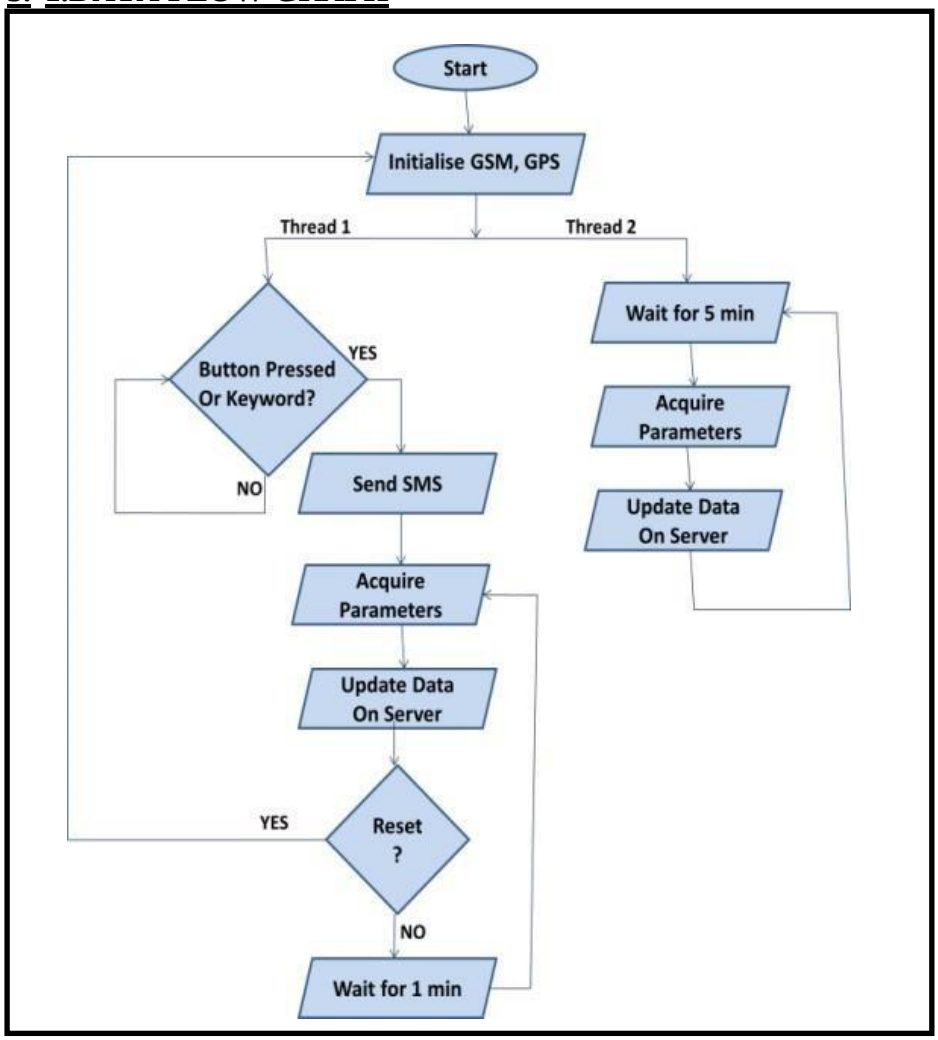
2) Extensive range monitoring system:

The application aside from conceding you to track down your children when they're within Bluetooth range, also functions when your kids go farther

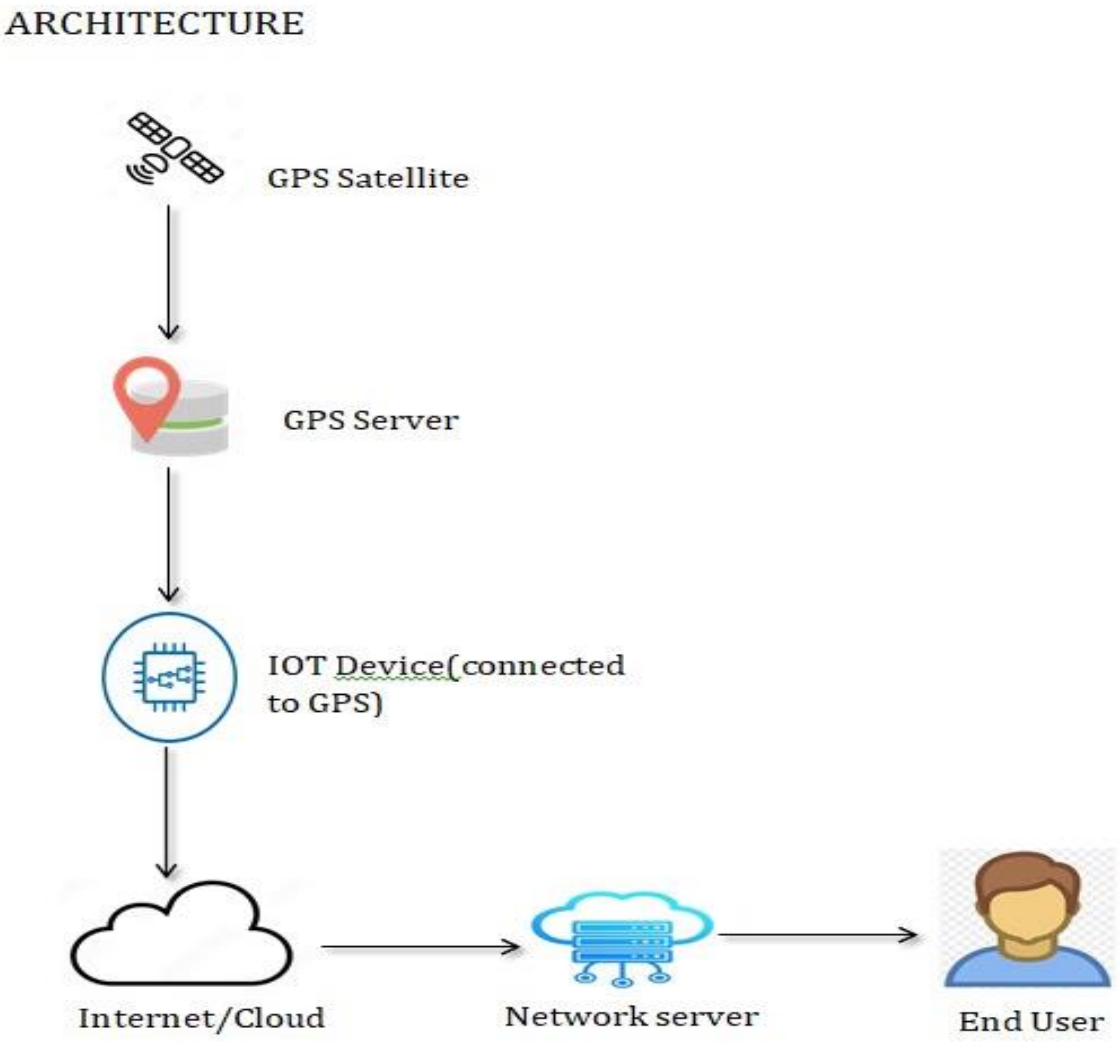
afield. Its competence as a tracker is outstanding if you live in densely populated areas like cities or big towns. This means you will be able to see the identity of the participating devices and it helps to diminish their vulnerability in harmful situations and also protects the children in emergency situations.

5.PROJECT DESIGN:

5. 1.DATA FLOW GRAPH

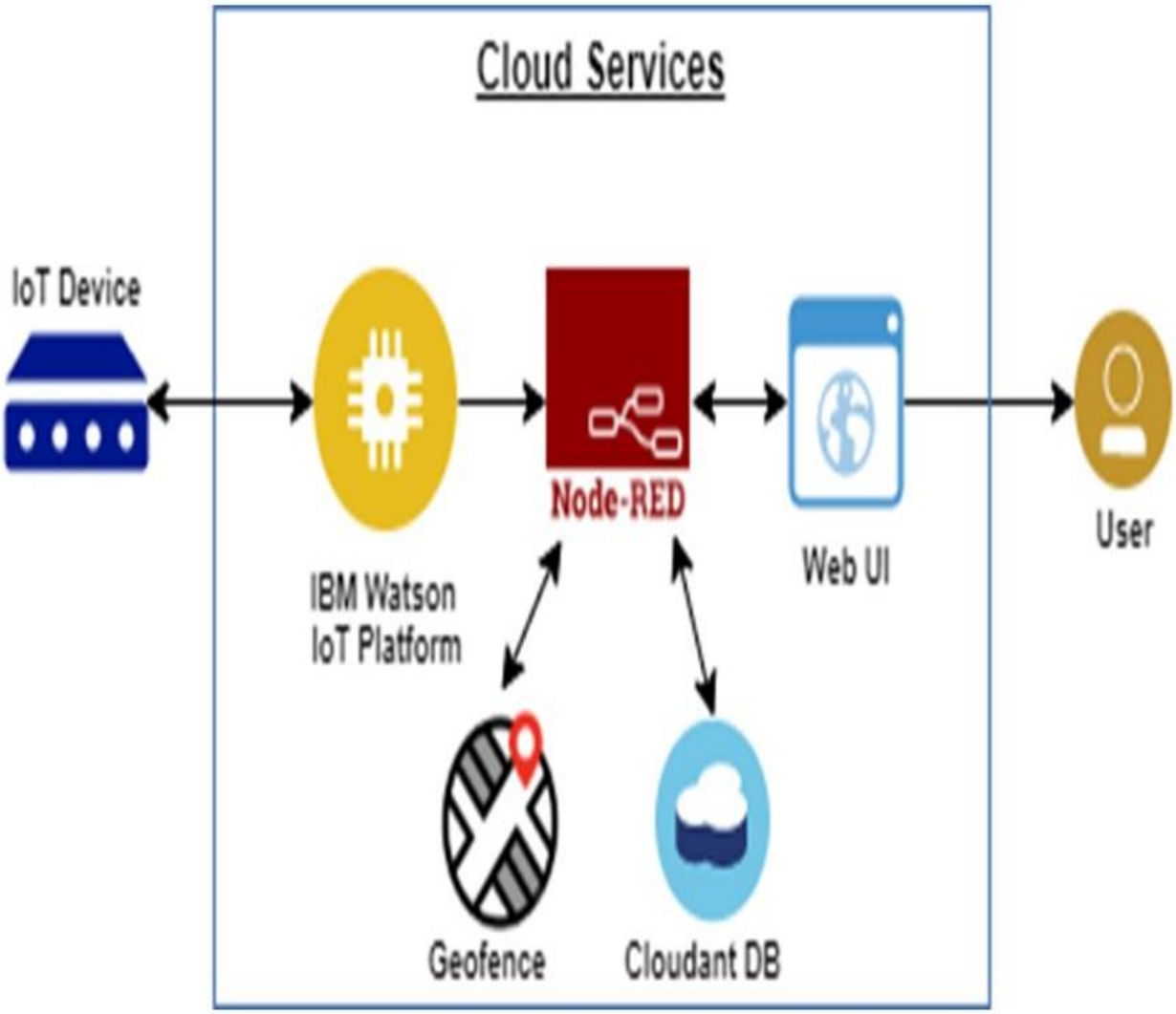


5.2. SOLUTION AND TECHNICAL ARCHITECTURE:



TECHNOLOGY ARCHITECTURE

ARCHITECTURE



5.4. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user) and (Web user)	Registration	USN-1	As a user, I can register my account by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered myself	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through apple account	I can register & access the dashboard with apple account Login	High	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering user id & password		High	Sprint-1
Customer Care Executive	Login		As I enter I can view the working of the application and scan for any glitches and monitor the operation and check if all the users are authorized.	I can login only with my provided credentials	Medium	Sprint - 3
Administrator	Login		Maintaining and making sure the database containing the locations are secure and accurate and updated constantly.	I can login only with my provided credentials	High	Sprint - 3

:

6. PROJECT PHASING AND SCHEDULING:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	Logesshwaran.R, Jeyaprakash.P
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application .	3	High	Jeron Jones.J, Azarudeen.N
Sprint-1		USN-4	As a user, I can register for the application	3	Medium	Logesshwaran Jeyaprakash.P
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	3	Low	Jeron Jones Azarudeen.N
Sprint-2		USN-5	As a user, I can logout of the application.	5	High	Logesshwaran Jeyaprakash.P
Sprint-4	Dashboard	USN-6	As a user, I can receive alert notifications if the movement is beyond the geofence.	13	High	Jeron Jones Azarudeen.N
Sprint-2		USN-7	As a user I can enter the coordinates and monitor the child's movement.	5	Medium	Logesshwaran Jeyaprakash.P
Sprint-3		USN-8	As a user I can update the coordinates whenever necessary.	13	Medium	Jeron Jones Azarudeen.N

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022	12	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	13	6 Days	14 Nov 2022	19 Nov 2022	13	19 Nov 2022

SPRINT PALNNING AND ESTIMATION

Sprint 1:

GEOFENCING CODE:

Basic Example Code: import time

```
def stopwatch(secon ds,d,lspoint): start = time.time()
            time.clock() elapsed
            = 0 flag = False num = 0
             while elapsed < seconds: elapsed       = time.time() - start print
                        "%02d" %
                        elapsed if elapsed > d[num] and elapsed < d[num+1] and flag == False: x =
                        lspoint[num][0] y =
                                    lspoint[num] [1] createpoint(x ,y) flag = True


print "Shot Taken"
print
                        point_in_poly(x,y,polygon) if elapsed > d[num+1]:
                                     print "Shot Taken" flag
== False
                                     num = num+1
x                                    =
lspoint[num]
[0]                                   y           =
                                    lspoint[num] [1] createpoint(x
                                    ,y) print
                        point_in_poly(x,y,polygon) time.sleep(1)


  def createpoint(x,y):
      crs =
    "point?crs=epsg:27700&field=id:i nteger" layer =
    QgsVectorLayer(crs, 'points' ,
    "memory")pr = layer.dataProvider()
```

```
      pt = QgsFeature() point1 =
    QgsPoint(x,y) pt.setGeometry(QgsGeometry.fromP
    oint(point1))pr.addFeatures([pt])
      # update extent of the layer layer.updateExtent s()
      # add the second pointpt
    = QgsFeature()
QgsMapLayerRegistry.instance().addMapLayers([layer])

def point_in_poly(x,y,poly):

      n =
    len(poly) inside =
    False

      p1x,p1y =
    poly[0] for i in range(n+1):
        p2x,p2y =
       poly[i % n]if y > min(p1y,p2y):

           if y <= max(p1y,p2y): if x <=
                max(p1x,p2x
                ):if p1y != p2y:
                   xints = (y-p1y)*(p2x-
                p1x)/(p2y-p1y)+p1xif p1x == p2x or x <= xints:
inside = not insidep1x,p1y =
                p2x,p2y
```

```
return inside


#### define the polygon polygon =

 [(512882.78819722467,120811.83924772343),(512960.8443717052 6,120809.7007223952),(512960.
  84437170526,120809.7007223952),(512959.77510904113,120754.0
 9906386107),(512882.78819722
  467,120756.2375891893)]


  #### set how long the script will run (70 seconds will get you in and out of geofence) time_seconds = 70

  #### first coordinatex =

 512915 y = 120728

  #### time intervals, 10 seconds between shots / or pointsintervals =

int(time_seconds / 10) lspoint = []

            #### build the list of coordinates to be plottedfor i

            in range(0,intervals+1): y1 = y + (i*12.5) lspoint.app

            end( [x,y1])


  #### to build the blocks of time in intervals, so we know the number of intervals (default is 7),

  #### we need a list of time intervals [0,10,20,30 etc] to check against the clock this list is d, f is thegap ie 10 seconds,
a is starting point (0)

  ### b is the number of intervals + 1 becuase the code will check the the next in the listf = 10

  a = 0 b = intervals+1

  d = [x * f for x in range(a, b)]


  ### Run the stopwatch, or start the program!

 stopwatch(time_seconds,d

 ,lsp oint)
```

Sprint 2:

Sprint 2 is about **LOGIN and NOTIFIACATION** of the IoT device in Parent's Web Application for getting information about Child's Status.

LOGIN:

This Coding is to built login page of parent's application to get information about child's condition.

Coding:

Sprint 2 is about LOGIN and NOTIFIACATION of the IoT device in Parent's Web Application for getting information about Child's Status.

LOGIN:

This Coding is to built login page of parent's application to get information about child's condition.

Coding:

```
<!DOCTYPE html>

<html> <head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Login Page </title>

<style>

Body { font-family: Calibri, Helvetica, sans- serif;background-color: #9FE2BF; }

button {

background-color: #9FE2BF;


width: 100%; color: black; padding: 15px; margin: 10px 0px;border: none; cursor: pointer; } form {

border: 3px solid #f1f1f1;


} input[type=text], input[type=password] {width: 100%; margin: 8px 0; padding: 12px 20px; display: inline- block;border: 2px white; box-sizing: border-box;


} button:hover {

opacity: 0.7;


}

.cancelbtn {
```

```
width: auto; padding: 10px 18px; margin: 10px 5px;
}

.container { padding: 25px; background-color: #CCCCFF;

}
</style> </head>
<body>
<center> <h1> Login Form </h1> </center>
<form>
<div class="container">
<label>Device ID/Number: </label>
<input type="password" placeholder="Enter Password" name="password" required>
<label>E-Mail : </label>
<input type="text" placeholder="Enter Username" name="username" required>
<label>Password : </label>
<input type="password" placeholder="Enter Password" name="password" required>
<button type="submit">Login</button>
<button class="loginBtn loginBtn--facebook">Login with Facebook.</button>
<button class="loginBtn loginBtn--google">Login with Google.</button>

<input type="checkbox" checked="checked"> Remember me
<button type="button" class="cancelbtn">
Cancel</button>Forgot <a href="#"> password? </a>
</div>
</form>
</body>
</html
```
NOTIFICATION:

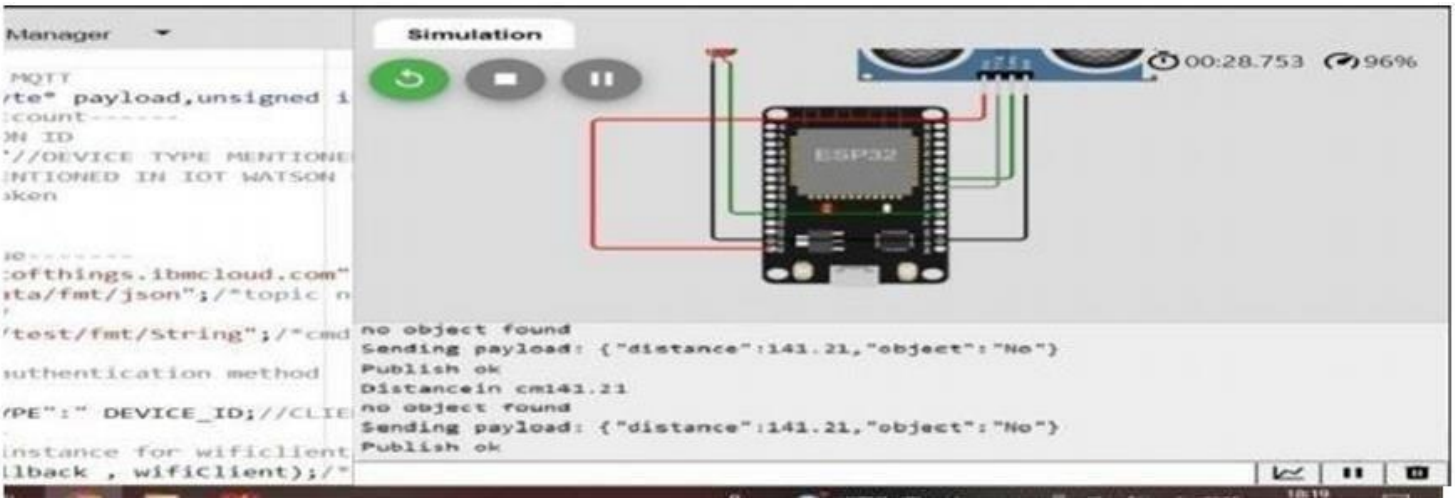
This coding will make connection between IoT Device & Parent's application. When the child crossacross the geofence message will be notified on parent's application. Coding:

```
#include<WiFi.h>//library for wifi

#include<PubSubClient.h>//library  for  MQTT

void  callback(char*  subscribetopic,  byte*  payload,unsigned  int  payloadlength);


//----------credentials  of  IBM  Account------


#define ORG "45z3o2"// IBM ORGANIZATION ID


#define  DEVICE_TYPE  "ESP32_Controller"//DEVICE  TYPE  MENTIONED  IN  IOT  WATSON

PLATFORM #define  DEVICE_ID  "bme2"//DEVICE  ID  MENTIONED  IN  IOT  WATSON  PLATEFORM #define TOKEN

"OKZ+q@JfPWDOd6wBTj"//TokenString data3;

float dist;


//------------customize the above value-------


char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name


char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event performand format in which data to be send*/

char  subscribetopic[]="ultrasonic/cmd/test/fmt/String";/*cmd  REPRESENT  Command

tupe and COMMAND IS TEST OF FORMAT STRING*/


char  authMethod[]="use-token-auth";//authentication  method

char token[]=TOKEN;

char  clientid[]="d:"  ORG  ":"  DEVICE_TYPE":"  DEVICE_ID;//CLIENT  ID


//
```

```
WiFiClient wifiClient;// creating an instance for wificlient


PubSubClient client(server, 1883 , callback , wifiClient);/*calling the predefined client idby passing parameter like server id,portand wificredential*/ int LED =4;


int trig

=5; int

echo=18; void setup(){

Serial.begin(115200)

;

pinMode(trig,OUTPUT

);

pinMode(echo,INPUT

);

pinMode(LED,OUTPUT) ; delay(10); Serial.println(); wificonnect(); mqttconnect();

}


void loop() {

digitalWrite(trig,LOW); digitalWrite(trig,HIGH); delayMicroseconds(10); digitalWrite(trig,LOW); float dur=pulseIn(echo,HIGH); float dist=(dur * 0.0343)/2; Serial.print("distance in cm");Serial.println(dist); PublishData(dist); delay(1000); if (!client.loop()){

mqttconnect(); }


}


/*.....retriving to cloud ..... */


void PublishData(float dist){

mqttconnect();//function call for connecting to ibm

/*creating the string in form of JSON to update the data to ibm cloud*/ String object;

if(dist<100)

{
```

```
digitalWrite(LED,HIGH); Serial.println("no object is near");object="Near";

}

else

{

digitalWrite(LED,LOW); Serial.println("no object found");object="No"; }

String payload="{\"distance\":"; payload +=dist; payload +="," "\"object\":\"";payload += object;

payload += "\"}";

Serial.print("Sending payload: ");

Serial.println(payload); if(client.publish(publishtopic, (char*) payload.c_str())){

Serial.println("Publish ok");/* if its sucessfully upload data on the cloud then it will print publish ok in serial monitor or else it will print publish failed*/ } else{

Serial.println("Publish failed");

}
```

Output:



Sprint 3:

LOCAL FORAGE:

!function(a)

{if("object"==typeof exports&&"undefined"!=typeof module)module.exports=a();else if("function"==typeof define&&define.amd)define([],a); else{var b; b="undefined"!=typeof window?window:"undefined"!=typeof global?global:"undefined"!=typeof

self?self:this,b.localforage=a()}}(function(){

return function a(b,c,d){

function e(g,h){if(!c[g]){if(!b[g]){

var i="function"==typeof require&&require; if(!h&&i)return i(g,!0);if(f)return f(g,!0); var j=new Error("Cannot find module '"+g+"'");

throw j.code="MODULE_NOT_FOUND",j}var k=c[g]={exports:{}};

b[g][0].call(k.exports,function(a){

var c=b[g][1][a];return e(c||a)},k,k.exports,a,b,c,d)}

return c[g].exports}

PNT2022TMID46033 for(var f="function"==typeof require&&require,g=0;g<d.length;g++)e(d[g]);

return e}({1:[function(a,b,c){(function(a){"use strict"; function c(){k=!0;for(var a,b,c=l.length;c;){

for(b=l,l=[],a=-1;++a<c;)b[a]();c=l.length}k=!1} function d(a){1!==l.push(a)||k||e()}var

e,f=a.MutationObserver||a.WebKitMutationObserver; if(f){var g=0,h=new

f(c),i=a.document.createTextNode("");h.observe(i,{characterData:!0}),e=function(){i.data=g=++g%2}} else if(a.setImmediate void ea.MessageChannel)e="document"in all"onreadystatechange in

a.document.createElement("script") function(){var b=a. document.createElement("script");b.onreadystatechange-function()(c(),b. onreadystatechange=null,b.parentNode.removeChild

(b),b=null),a.document.documentElement.appendChild(b)}:function() (setTimeout(c,8));else{var j-new

a.MessageChannel;j.port1.onmessage=c,e-function(){j.port2.postMessage(0)}}var k,1-[];b.exports- d)).call(this, "undefined"!=typeof global?global: "undefined"!=typeof self?self: "undefined"!=typeof window?window:

{})},{}],2: [function(a,b,c){"use strict"; function d()() function e(a){if("function" I-typeofa) throw new

TypeError("resolver must be a function"); this.states, this.queue=[], this.outcome vald

0,aldi(this,a)} function f(a,b,c){this.promise-a, "function"==typeof b&&(this.onFulfilled-b, this.callFulfilled-this.otherCallFulfilled), "function"typeof c&&(this.onRejected=c,

INDEX:

```
<!DOCTYPE html>

<html lang="en" style="height: 100%; margin: 0;">

<head>

<meta charset="UTF-8" />

<meta name="description" content="The Home Page after Logged In" />


<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>IOT Based Safety Gadget for Child Safety Monitoring and Notification</title>

<script src="./localforage.js"></script>

<script> if (window.location.hostname !== "localhost") {if

(location.protocol !== "https:") { location.replace(

`https:${location.href.substring( location.protocol.length

)}`

)

}

}
```

```
async function check() {
let data = localforage.getItem("userData")if
(data == null) { window.location.href = "/login"
} } check()
</script>
</head>
<body
style=" height: 100%; margin: 0;         font-weight: 300;
 font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,  Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
       sans-serif;
"
>
<div  class="wrapper" style=" height: 90%;display: flex;
flex-direction: column; align-items: center; justify-content: center;
text-align: center;"
>
<div  class="details" style=" display: flex; flex-direction: column; align-items: center; gap:
20px; padding: 1rem; border-radius: 5px; box-shadow: 0 0 8px 0px #44444444; max-width: 80%;
"
>
<h1 class="name" style="margin: 0"></h1>
<div class="imageContainer"
style="padding: 10px; height: 10rem; width: 10rem"
>
<img class="image" alt="profile picture" />
</div>
<h2 class="email" style="margin: 0"></h2>

<a style="text-decoration: none;text-align: center;font-size: 1.2rem;color: #0070f3;font-weight:400;" href="./dashboard">Go to Dashboard ?</a>
```

```
</div>

</div>

<script> async function main() {

let name = document.querySelector(".name") let image = document.querySelector(".image")let email =
document.querySelector(".email") let userData = await localforage.getItem("userData") if(userData == null) {
window.location.href = "/login"

}

name.innerHTML = `Welcome ${userData.firstName} ${userData.lastName}!` image.src = userData.profilePic

email.innerHTML = `Your email is: <a style="text-decoration: none;color: #0072B5;"

href="mailto:${userData.email}">${userData.email}</a>`

} main()

</script>

</body> </html>
```

Sprint 4:

FIREOAUTH:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="/css/fireoauth.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.css">
<link rel="shortcut icon" href="https://raw.githubusercontent.com/tharunoptimus-pd/firepwa/main/favicon.ico?token=GHSAT0AAAAAABR46HVJ5M5L3QGFRZRQXOISYUJU WAA" type="image/x-icon">
<style> html, body {
height: 100%; margin: 0;
font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen,
Ubuntu, Cantarell, "Open Sans", "Helvetica Neue", sans-serif; font-weight: 300;
}

a {
text-decoration: none; color: #007bff; font-weight: 500; font-size: 1.2rem;
}

h3 {
font-size: 1.4rem;
} h3, h4 { margin: 0;
padding: 0.3rem 0;
}
.wrapper { display: flex;
flex-direction: column; align-items: center; justify-content: center; height: 100%; text-align: center;
}

.oneClickSignin { padding: 0.5rem; border: 1px solid #44444444; border-radius: 5px; box-shadow: 0 0 3px 0px #44444444;
opacity: 0.2; pointer-events: none;
} .qrcode { opacity: 0.1; } .learnAboutFire { padding-top: 1.25em;
}
```

```
.qrHolder { display: none; margin-top: 3rem;
}

.qrContainer { align-items: center; display: flex; justify-content: center; padding: 8px; margin: 2rem auto; box-shadow: 0 0px 6px 1px rgb(0 0 0 / 16%); border: 1px solid #44444444; border-radius: 6px; width: 200px; height: 200px;
}
</style>
<title>Fire OAuth</title>
<script> if (window.location.hostname !== "localhost") {
if (location.protocol !== "https:") { location.replace(
`https:${location.href.substring( location.protocol.length
)}`
)
}
}
</script>
</head>
<body>
<div class="wrapper">
<h3 class="pageTitle">Login with Fire ??</h3>
<div class="qrAuthorize">
<h4 class="subTitle">Scan QR from your Fire OAuth App??</h4>

<div class="qrContainer">
<canvas id="qr-code" class="qrcode"></canvas> </div>
</div>

<div class="oneClickSignin">
<h4>Have Fire PWA on this device?</h4>
<a target="_blank" id="authorizeOverLink" href="https://firepwa.netlify.app/authorize?sessionId" rel="noopener">Click to Authorize ?? </a> </div>

<div class="learnAboutFire">
<a target="_blank" href="https://fireoauth.netlify.app" rel="noopener">Learn More about Fire ??</a>
</div>
</div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.js"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/qrious/4.0.2/qrious.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.2.0/socket.io.js"></script>
<script> const FIRE_API_KEY = "635b790a3bcc6b59c4b772d0"
const FIRE_ENDPOINT = "https://fire.adaptable.app/api/apis/generate" const CHANNEL_NAME = "fireOAuthChannel"
const broadCastingChannel = new BroadcastChannel(CHANNEL_NAME)
const FIRE_SERVER_SOCKET_ENDPOINT =
"https://fire.adaptable.app"let socket =
io(FIRE_SERVER_SOCKET_ENDPOINT) let qr let qrcode = document.querySelector(".qrcode")
let oneClickSignin = document.querySelector(".oneClickSignin") let pageTitle = document.querySelector(".pageTitle") let subTitle = document.querySelector(".subTitle")

function setOpacity(opacity) {
oneClickSignin.style.opacity = opacity
oneClickSignin.style.pointerEvents = opacity === "1" ? "auto" : "none" qrcode.style.opacity = opacity
}

async function getSessionID()
{let response try { response = await
fetch(`${FIRE_ENDPOINT}/${FIRE_API_KEY}`, {method:
"GET", headers: {
"Content-Type": "application/json",
}
})
} catch (error) { console.log(error) return null
}

let data = await response.json() let { sessionId, chatRoomId } = data return { sessionId, chatRoomId } }

function generateQR(value) {
(qr = new QRious({ element: document.getElementById("qr-code"), size: 200, level: 'M', value: value,
}))
}

function changeHREF ({sessionId, chatRoomId}) { let firePwaUrlHostname = "https://firepwa.netlify.app" let originURL = encodeURIComponent(window.location.origin)
 let url =
```

```
`${firePwaUrlHostname}/authorize.html?sessionId=${sessionId}&chatRoomId=${chatRoomId}&url=${ori ginURL}` let a = document.getElementById("authorizeOverLink") a.href = url
}
async function fire() { NProgress.set(0.4)
let { sessionId, chatRoomId } = await getSessionID()

        if(sessionId === undefined || chatRoomId === undefined || sessionId === null || chatRoomId === null)
{       pageTitle.innerHTML = "Something went wrong ???" subTitle.innerHTML = "Please try again later ????" return
}
 setOpacity("1")


NProgress.done() let data = { sessionId,
url: encodeURIComponent(window.location.origin)
}
data = JSON.stringify(data) generateQR(data) changeHREF({sessionId, chatRoomId})socket.emit("join room",
sessionId)
}

fire()
socket.on("trusted token", (token) => {

let data = {} data.success = true data.token = token

broadCastingChannel.postMessage(data)

window.close()
})
</script>
</body>
</html>
```

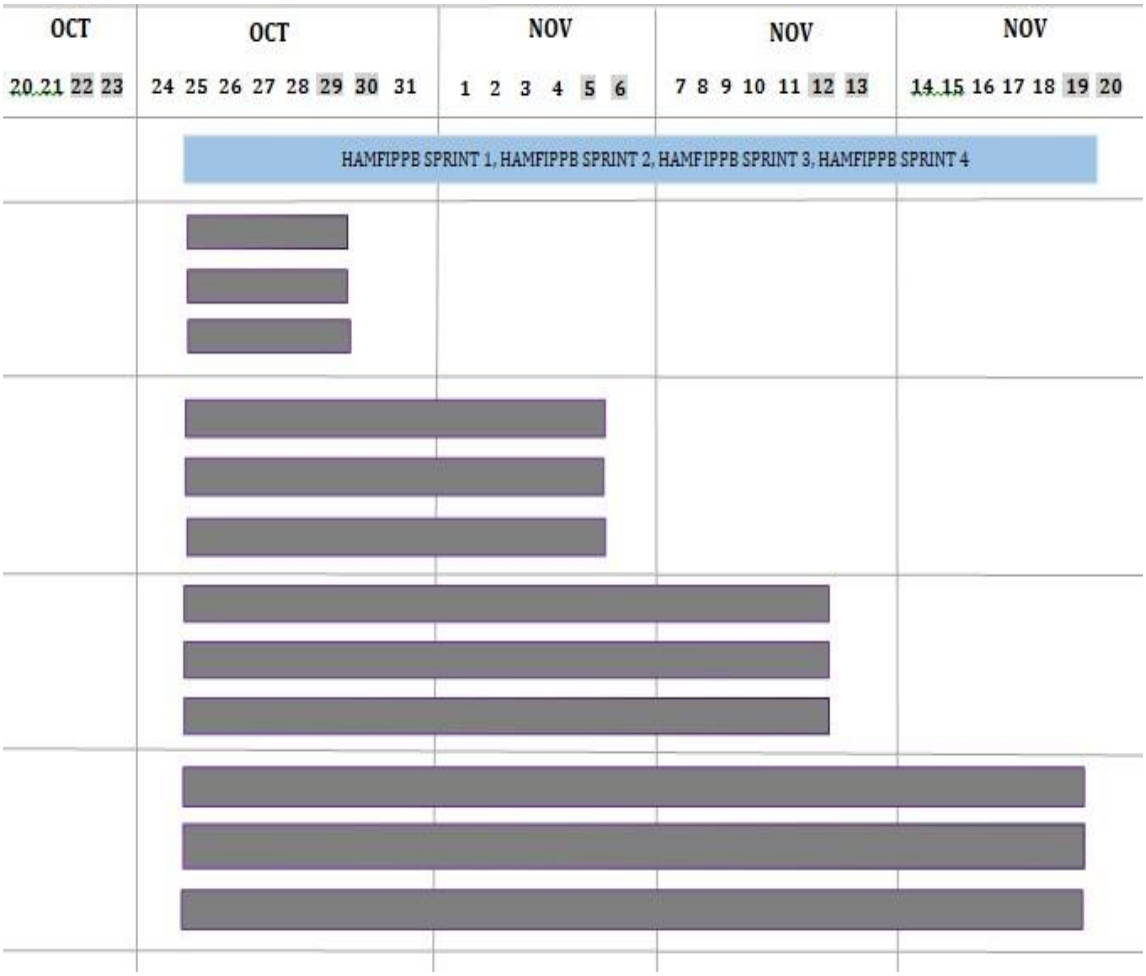
DASHBOARD:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="./css/dashboard.css">
<title>Dashboard</title>
<script src="./localforage.js"></script>
</head>
<body>
<div class="wrapper">
<div class="header">
<span class="heading">Dashboard</span>
<span class="right">
<span class="username">Hello User</span>
<span>
<img class="profilePic" src="https://avatars.dicebear.com/api/avataaars/asdfasdfds.svg"
alt="User Profile" height="30" width="30">
</span>
</span>
</div>

<div class="actionCenter">
<div class="action">
<span>Create Child Card</span>
</div>
<div class="action">
<span class="logout">Log out</span>
</div>
</div>

<div class="childCardContainer">
<div class="childCard">
<div class="childCardHeader">
<span>Child Name</span>
<span>Age 12</span>
</div>
<div class="actions">
<span>View</span>
```

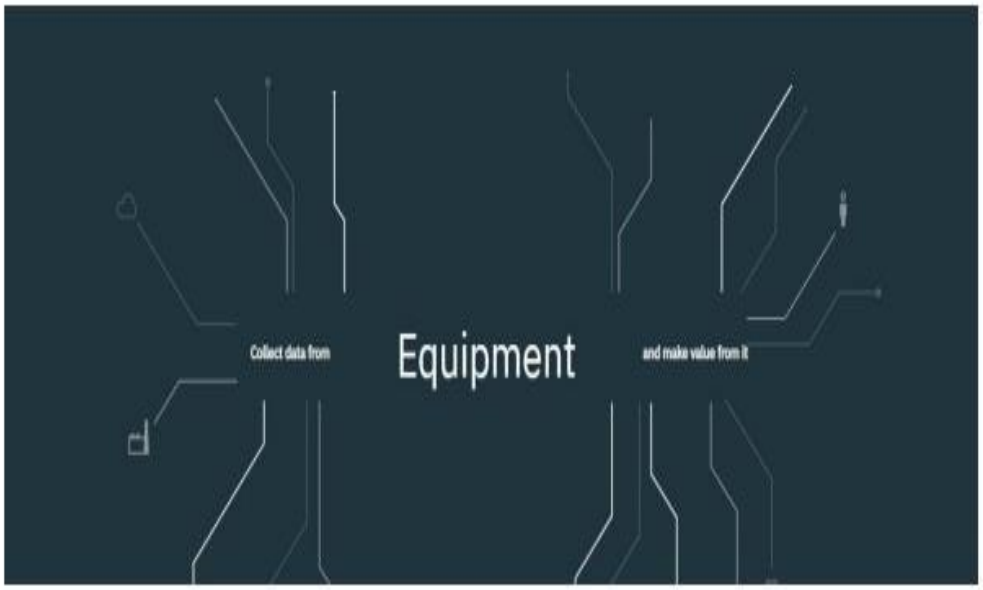
```
<span>GeoFence</span>
</div>
</div>
</div>
</div>
<script> async function main() { let userData = await localforage.getItem('userData') if(userData == null) {
window.location.href = "/login"
}
document.querySelector(".username").innerHTML = `Hello ${userData.firstName}`
document.querySelector(".profilePic").src = userData.profilePic
} main()
document.querySelector(".logout").addEventListener("click", async () => {
await localforage.setItem('userData', null) window.location.href = "/login"
})
</script>
</body>
</html>
```

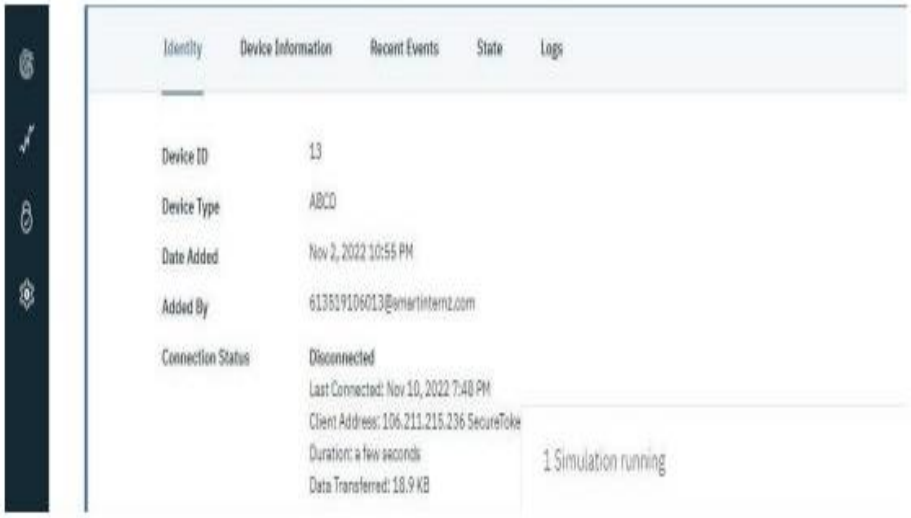
BURNDOWN GRAPH:


OCT
OCT
NOV
NOV
NOV
20 21 22 23
24 25 26 27 28 29 30 31
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
HAMFIPPB SPRINT 1, HAMFIPPB SPRINT 2, HAMFIPPB SPRINT 3, HAMFIPPB SPRINT 4

7. CODING:

Creating IBM Cloud Service and creating the device:



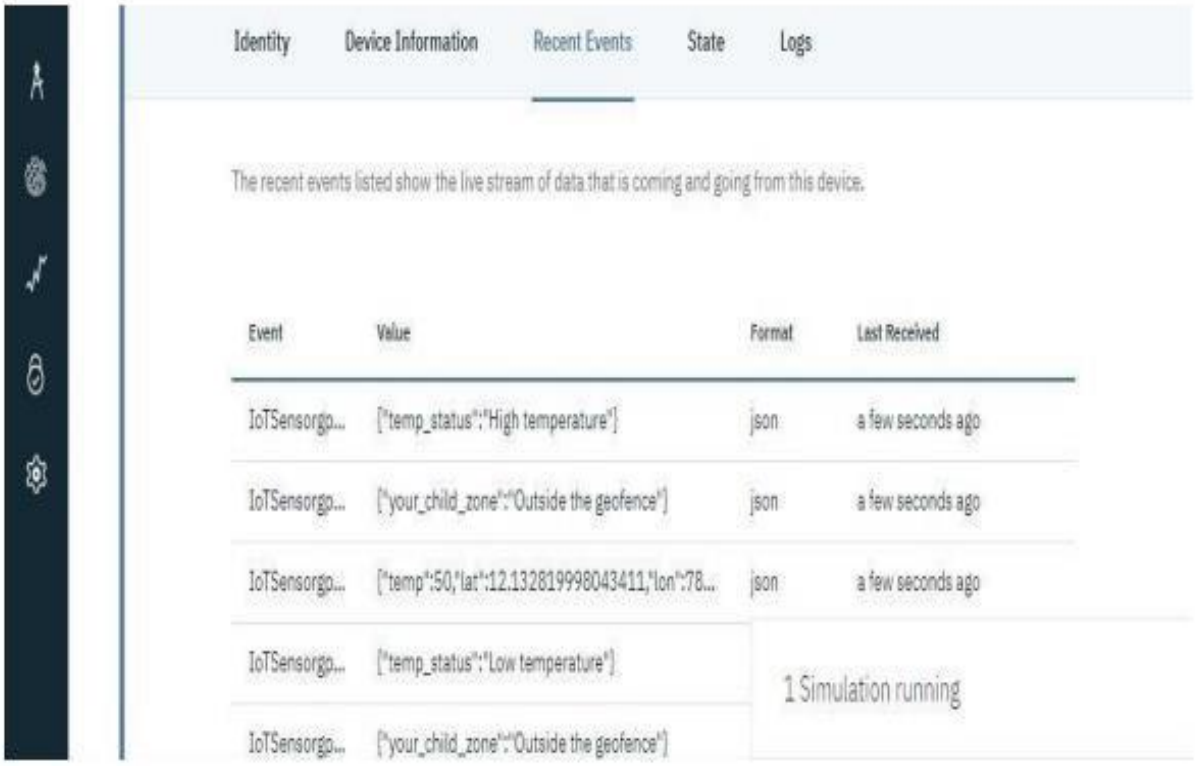


Utilization and Optimization of Python Code:

```
import timeimport
sys
import ibmiotf.applicationimport
ibmiotf.device import random
#Provide your IBM Watson Device Credentialsorganization = "zwx6lb"
deviceType = "ABCD" deviceId = "13"
authMethod = "token" authToken =
"12345678"
#api key {a-illza1-mbdxqo6z0s} #api token
{zSYzlSuAWF&F_x7GkT}

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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type"greeting" 10 times
print("power on ")
print("checking connection to waston iot...")time.sleep(2)
deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")
print("i can provide your children live location and temperature ")print()
name=str(input("enter your child name:"))while True:

    temperature=random.randint(20,50)#random temperature for your child latitude=random.uniform(10.781377,10.78643)#random
    latitude for your child longitude=random.uniform(79.129113,79.134014)#random longitude for your childa="Child inside the geofence"
    b=" Child outside the geofence"c="High
    temperature"
    d="Low temperature"
    x={'your_child_Zone':a}
    y={'your_child_Zone':b}
    z={'temp_condition':c}
    w={'temp_condition':d}

    data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name  }#print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temperature, "latitude = %s %%" % latitude,"longitude = %s %%" % longitude, "to IBM Watson")
        print("\n")
    success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,on_publish=myOnPublishCallback)

    if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and longitude
<=79.133000:
```

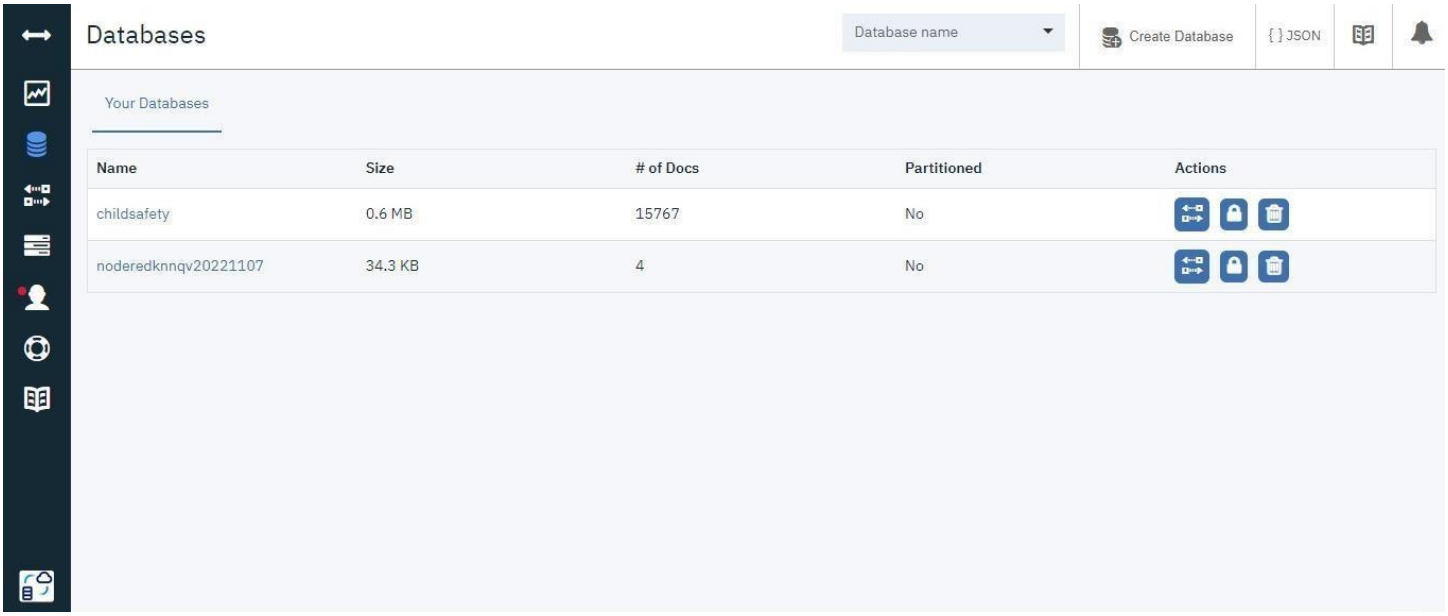
```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=myOnPublishCallb ack)

            print(x) print("\n")

        else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublishCallb ack)

            print(y) print("\n")

        if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallb ack)

                print(c) print("\n")

        else:

        deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCall back)

                print(d) print("\n")

        if not success:

            print("Not connected to IoTF")print("\n")

        time.sleep(3)

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

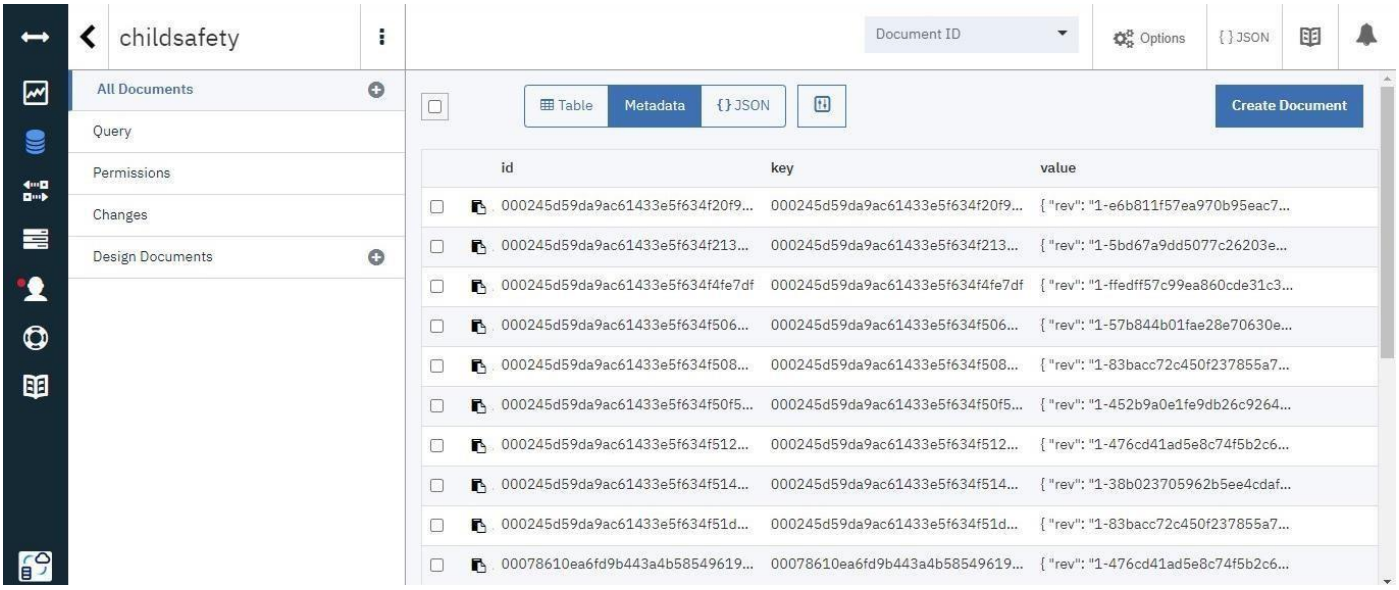
Connecting IBM Watson and python Code Debugging and Traceability:



8. TESTING:

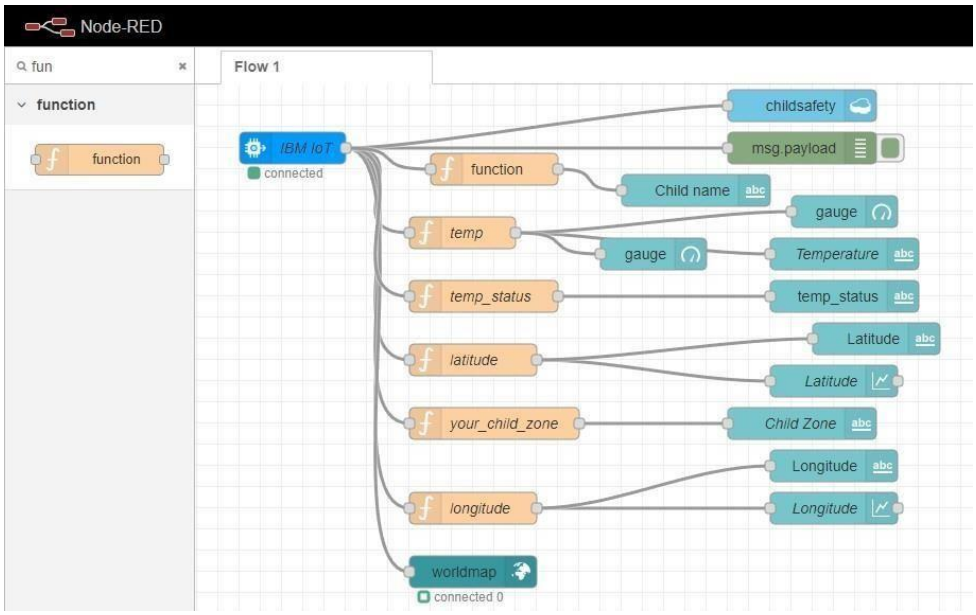
8.1. Create Cloudant DB:





8.2.UTILIZATION OF TESTING TOOLS:

1)Node-Red Service with Cloudant DB:



2) APP Inventor:

To monitor the children continuously according to their surroundings and movability.

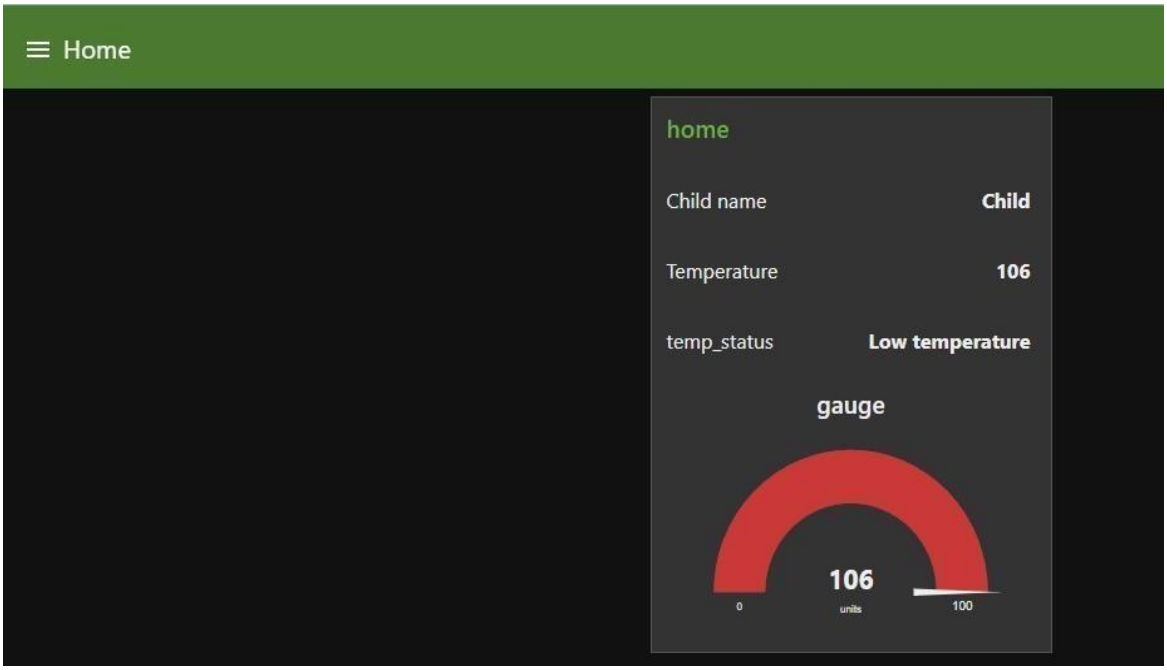
Monitor accordingly to their environments and their circumstances and alert the system and the moniter.

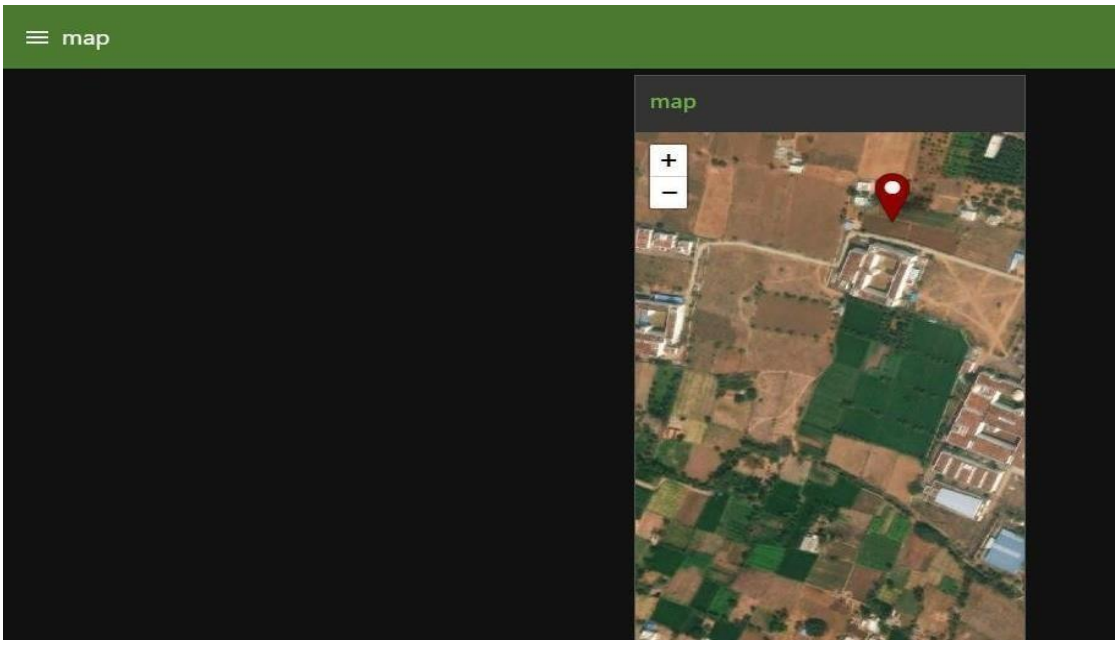
9.RESULT:

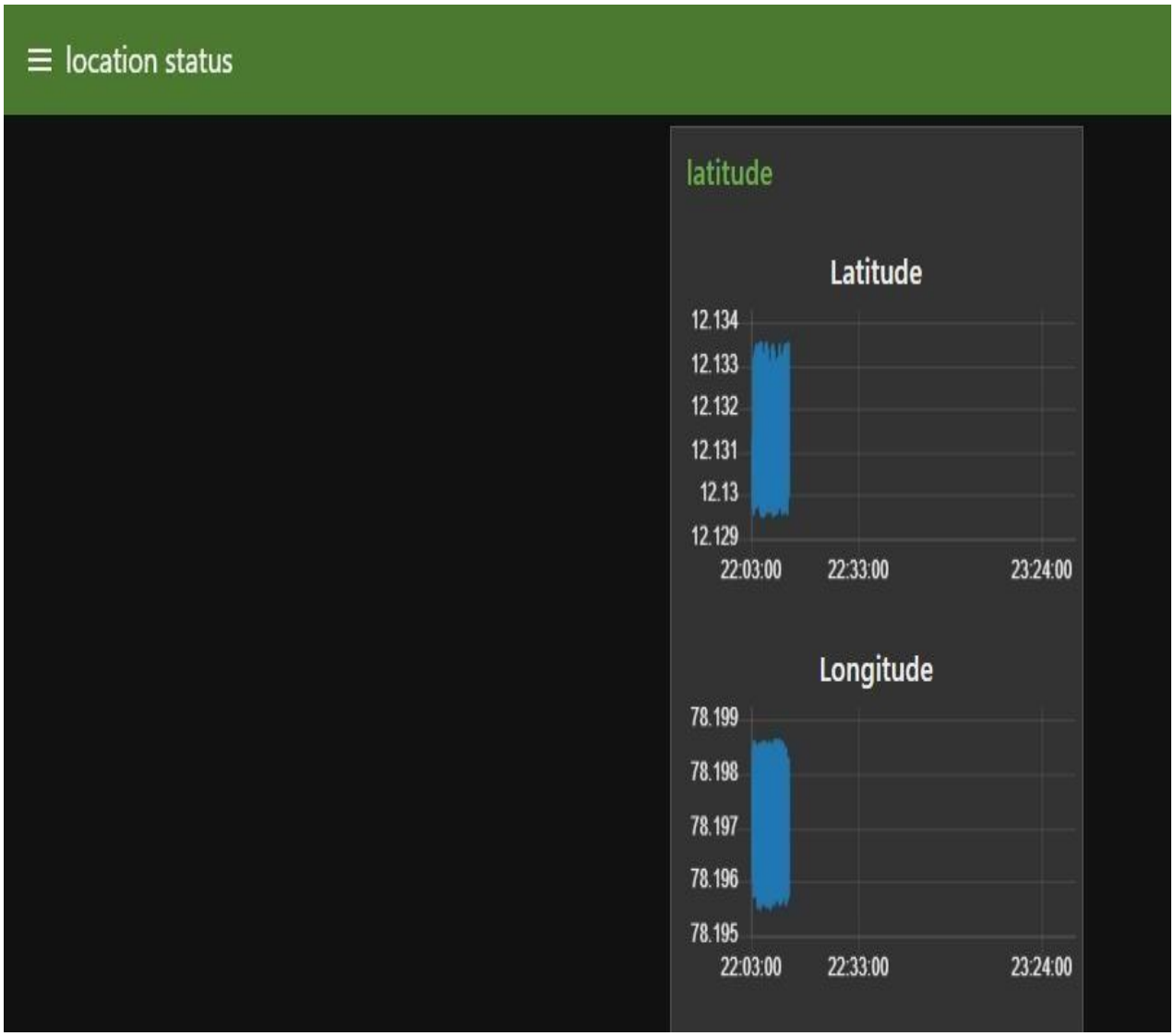
9.1. PERFORMANCE ANALYSIS:

APPLICATION PERFORMANCE METRICS:

WED UI:







10. ADVANTAGES:

- Wearable device is that, according to its design, it can be accessed from any mobile device and does not mandate a lot of technical knowledge from the user to operate.

- The device has IoT monitoring that allows the child to be monitored.
- It also has numerous sensors that are used to detect exact signals such as heart rate, temperature, and other dangers and alert the parents.

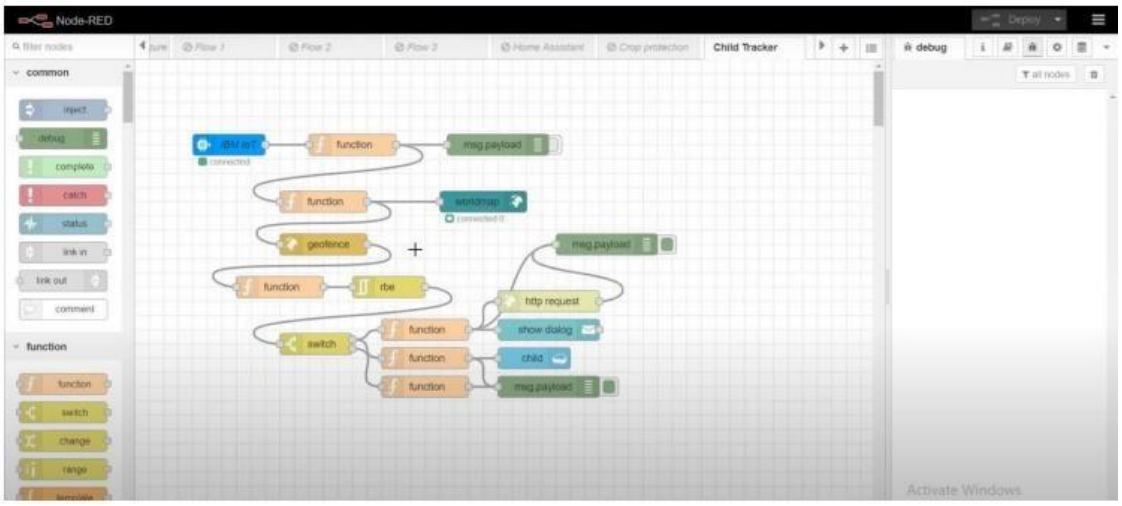
11.CONCLUSION:

This paper surveys various papers related to an IOT based safety wearable device that helps the parents or guardians to monitor the safety of their ward or children. The main aim is to provide an effective and convenient solution to the parents or guardians to keep track of their child's safety.

In summary, the parents or guardians will be alerted if abnormal values are read by the sensor or if values on these sensors cross a given threshold value, alerting them that the child could be in danger. This helps the parents to locate and monitor their child's safety.

12.APPENDIX:

SOURCE CODE:




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

myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "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()
```

13. FUTURE SCOPE:

This system can be further enhanced by installation of mini camera inside smart gadget for better security so that live footage can be seen on parental phone during panic situations. The system can be modified by installation of small solar panels for charging the battery of smart gadget to gain maximum battery backup.

14.GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-49690-1660834871

15:DEMO LINK:

https://youtu.be/fUmNeeKCthA

CHAPTER 14
REFERENCE

[1] Arun Francis G, Janani I, Kavya S and Ramiyadevi K. Child Safety Wearable Device Using Raspberry Pi. Waffen-UND Kostumkunde Journal. 11(2). 2020. pp.135-137.

[2] A. Helen, Kalaiselvi V.K.G, M. Fathima Fathila and R. Rijwana. A smart watch for women security based on iot concept 'watch me', International Conference on Computing and Communications Technologies (ICCCT). 2017.

[3] Alexey Vinel Feng Xia and Laurence T. Yang and Lizhe Wang. Internet of Things. International Journal of Communication Systems. 25(9). 2012. pp.1101-1102. DOI: https://doi.org/10.1002/dac.2417

[4] Anjum Khairi, M.U. Farooq, Muhammad Waseem, Sadia Mazhar and Talha Kamal, M.U. Farooq, Muhammad Waseem and Sadia Mazhar. A Review on Internet of Things (IoT). International Journal of Computer Applications. 113(1). 2015. pp.1-7. DOI: https;//doi.org/10.5120/19787-1571

[5] Arun K Mani1, M.Gokilavani, Shreevani D, Samra Said and Unnikrishnan K N. A Review: IoT And Cloud Computing For Future Internet. International Research Journal of Engineering and Technology (IRJET). 6(5). 2019. pp.1098-1102.

[6] AbdelRahman H. Hussein. Internet of Things (IOT): Research Challenges and Future Applications. (IJACSA) International Journal of Advanced Computer Science and Applications. 10(6). 2019. pp77-82.

[7] Chamandeep Kaur. The Cloud Computing and Internet of Things (IoT). International Journal of Scientific Research in Science, Engineering and Technology. 7(1). 2020. pp.19-22. DOI: https://doi.org/10.32628/IJSRSET196657

[8] Cohen D and Crabtree B. RWJF - Qualitative Research Guidelines Project | Semi-Structured Interviews | Semi-Structured Interviews. Qualres.org 2008.

[9] D. Ezhilarasi, N. Senthamilarasi Bharathi and R.B. Sangavi. Child Safety Monitoring System Based on IoT. Journal of Physics: Conference Series. 1362(1). 2019. pp.1742-6596.

[10] Dr. J. Jegathesh Amalraj, J. Jereena John and S. Banumathi. IOT Sensors And Applications: A Survey. International Journal Of Scientific & Technology Research. 8(8). 2019. pp.998-1003.

[11] Dr. R. Nagaraja and P. Elamathi. Smart Children Safety Using Wearable Device - A Review. International Journal of Advanced Research in Electrical,

Electronics and Instrumentation Engineering. 8(11). 2019. pp.2278-8875.
[12] Dr. R. Rajesh and Sureshkumar P.H. The Analysis of Different Types of IoT Sensors and security trend as Quantum chip for Smart City Management. IOSR Journal of Business and Management (IOSR-JBM). 20(1). 2018. pp.55-60. DOI: https://doi.org/10.9790/487X-2001045560
[13] Dissertation.laerd.com. Quota Sampling | Lærd Dissertation. 2021.
[14] E Kusuma Kumari, K N H Srinivas, M Nandini Priyanka, S Murugan and T D S Sarveswararao. Smart IOT Device for Child Safety and Tracking. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 8(8). 2019. pp.2278-3075.
[15] Fadi Muheidat, Lo'ai Tawalbeh, Mais Tawalbeh and Muhannad Quwaider. IoT Privacy and Security: Challenges and Solutions. Applied Sciences. 10(12). 2020. p.4102. DOI: https://doi.org/10.3390/app10124102
[16] Hanan M. Shukur, Lailan M. Haji, Mohammad A. M. Sadeeq, Omar M. Ahmed, Rizgar R. Zebari and Shakir M. Abas. Journal of Applied Science and Technology Trends. 1(2). (2019). pp40-47.
[17] Jin-kuang Wang, Jie-lun Li, Jing Zhang, Min-feng Yao and Shi-Xing Li. Research and Application of Internet of Things. Journal of Machine to Machine Communications. 1(3). 2015. pp.215-228. DOI: https://doi.org/10.13052/jmmc2246-137X.132
[18] Mohsin Nazir. Cloud Computing: Overview & Current Research Challenges. IOSR Journal of Computer Engineering. 8(1). 2012. pp.14-22.
[19] Nishit Raghuwanshi Rudra. 8 Best GPS Trackers For Kids In 2020 To Ensure They Are Safe. 2020.
[20] Libguides.wits.ac.za. 2020. Libguides: Research Support: Research Methodology.
Atlantis Highlights in Computer Sciences, volume 4
471
[21] Paul J. Lavrakas. Encyclopedia of Survey Research Methods. 2008. DOI: httpss://doi.org/ 10.4135/9781412963947.n424
[22] QuestionPro. Online Surveys: Definition, Characteristics, Examples, Advantages And Disadvantages | Questionpro. 2020.
[23] Robert J. Shapiro, Kevin A. Hassett. The Economic Benefits Of Reducing Violent Crime - Center For American Progress. 2012.
[24] Saul McLeod. Questionnaire: Definition, Examples, Design and Types. 2018.

[25] Tom Pollock. The Difference Between Structured, Unstructured & SemiStructured Interviews - Oliver Parks Consulting LLC - Technology Sector Recruitment Experts. 2020.
[26] United Nations: Office on Drugs and Crime. Crime Prevention. 2020.
[27] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, T. Sato, “Secure Artificial Intelligence of Things for Implicit Group Recommendations”, IEEE Internet of Things Journal, 2021, doi: 10.1109/JIOT.2021.3079574.
[28] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin and G. Srivastava, "An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things," IEEE Journal of Biomedical and Health Informatics, 2021, doi: 10.1109/JBHI.2021.3075995.
[29] L. Tan, K. Yu, F. Ming, X. Cheng, G. Srivastava, “Secure and Resilient Artificial Intelligence of Things: a HoneyNet Approach for Threat Detection and Situational Awareness”, IEEE Consumer Electronics Magazine, 2021, doi: 10.1109/MCE.2021.3081874.
[30] B D, Parameshachari & Rachana, C. (2017). CLOUD COMPUTING: A RESEARCH PERSPECTIVE ON THE SECURITY ISSUES.
[31] Rachana, C.R., Banu, R., Ahammed, G.A. and Parameshachari, B.D., 2017, August. Cloud