Project	IoT Based Safety Gadget for Child Safety Monitoring and Notification
Team Id	PNT2022TMID19513
Member1	Harish S
Member2	Disha D
Member3	Durga Devi S
Member4	Yuvan Sanjay K

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Taking care of a baby is always a hectic problem and also maintaining a busy schedule life and doing a job, leaving a child with someone else cannot assure your safety. They can never assure themselves at work that their baby will be safe or baby is being properly cared for by the maid or caretaker. To remove all these problems and to make a feasible and successful application so that parents can have a track of their children.

1.2 PURPOSE

Child safety and tracking is a major concern as the more 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. Actually, there is a need to use IoT-based child security system since the safety of children has become a major concern. In fact, crimes on children keep increasing despite actions have been taken by the government.

The overall percentage of child abasements worldwide is about 80% nowadays, out of which 74% are girls and the remaining are boys. For every 40 seconds, a child is gone missing in the world. Due to that, parents are worried for their children and perhaps, a hard challenge for them to guarantee safety of their children when they are out. IoT is applied to propose a wearable smart band which helps parents to monitor and get known of their child's condition at anywhere and anytime even if they are not by their children side.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Child Safety Monitoring System Based on IOT

The overall percentage of child abusements filed nowadays in the world is about 80%, out of which 74% are girl children and the rest are boys. For every 40 seconds, a child goes missing in this world. Children are the backbone of one's nation, if the future of children was affected, it would impact the entire growth of that nation. Due to the abusements, the emotional and mental stability of the children gets affected which in turn ruins their career and future.

These innocent children are not responsible for what happens to them. So, parents are responsible for taking care of their own children. But, due to economic condition and aims to focus on their child's future and career, parents are forced to crave for money. Hence, it becomes difficult to cling on to their children all the time. In our system, we provide an environment where this problem can be resolved in an efficient manner. It makes parents to easily monitor their children in real time just like staying beside them as well as focusing on their own career without any manual intervention.

Baby Monitoring System using Image Processing and IoT

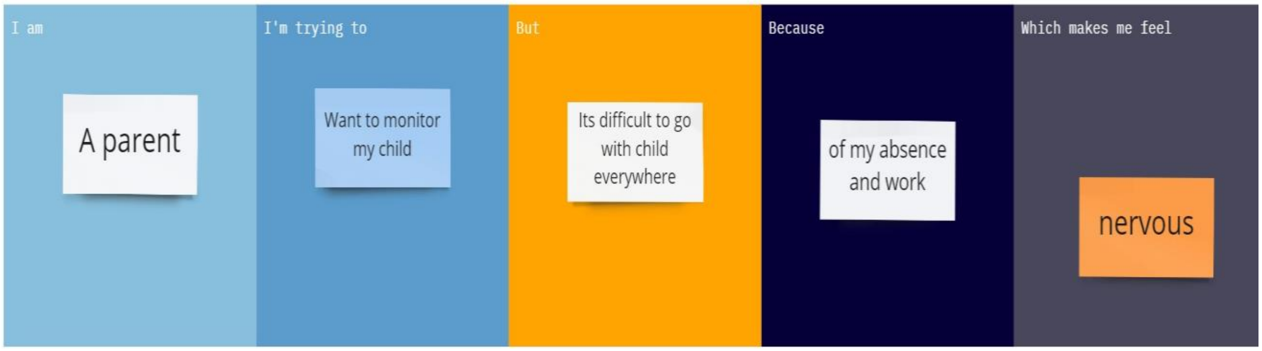
Non-contact-based baby monitoring system using image processing is proposed in this paper which is used for proper safety and monitoring the activity of baby by their busy parents. The system detects the motion, crying and present position of the baby. If any abnormal action is detected, then the system sends a message in the form of text and images of baby to the particular user through email. Raspberry Pi B+ module is used to process the videos taken by pi camera, MIC is used for crying detection and image processing is used for detection of real-time motion of babies and boundary condition of the bed. The system required to first install OS Raspbian, and all the other packages like OpenCV, NumPy and Virtual environment.

Face detection algorithm is trained using Haar classifier for positive face images and negative nonface images. This system will help in decreasing the chances of the baby's falling from the bed. Also, this system can be used in hospitals while baby is sleeping where the stress among the nurses will be reduced.

2.2 REFERENCES

1. N. Senthamilarasi, N. Divya Bharathi, D. Ezhilarasi and R.B. Sangavi "Child Safety Monitoring System Based on IoT" Published under licence by IOP Publishing Ltd.
2. Yogita K. Dubey, Sachin Damke "Baby Monitoring System using Image Processing and IoT" September 2019 International Journal of Engineering and Advanced Technology.

2.3 PROBLEM STATEMENT DEFINITION



I am

A parent

I'm trying to

Want to monitor my child

But

Its difficult to go with child everywhere

Because

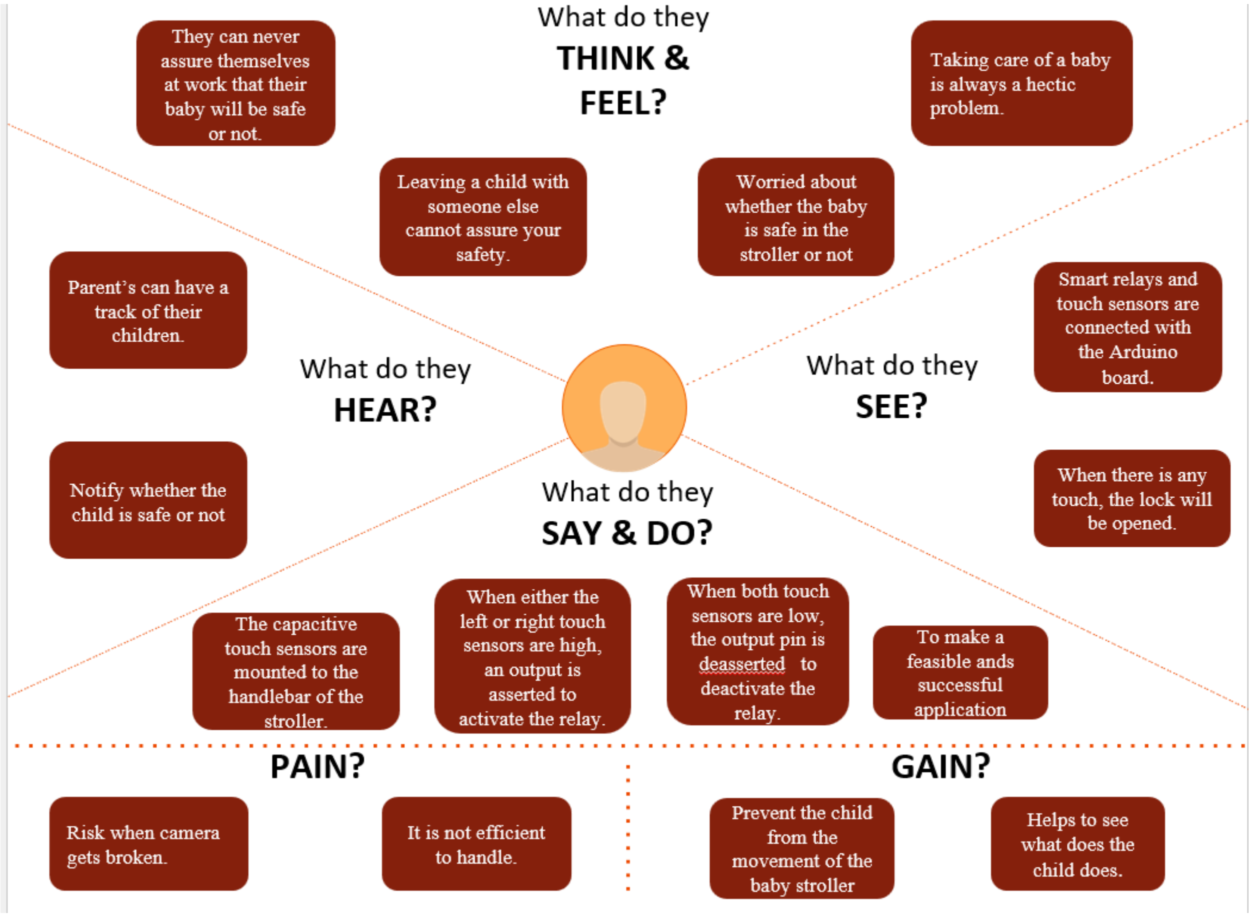
of my absence and work

Which makes me feel

nervous

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



What do they THINK & FEEL?

They can never assure themselves at work that their baby will be safe or not.

Taking care of a baby is always a hectic problem.

Leaving a child with someone else cannot assure your safety.

Worried about whether the baby is safe in the stroller or not

What do they HEAR?

Parent's can have a track of their children.

Notify whether the child is safe or not

What do they SEE?

Smart relays and touch sensors are connected with the Arduino board.

When there is any touch, the lock will be opened.

What do they SAY & DO?

The capacitive touch sensors are mounted to the handlebar of the stroller.

When either the left or right touch sensors are high, an output is asserted to activate the relay.

When both touch sensors are low, the output pin is deasserted to deactivate the relay.

To make a feasible ands successful application

PAIN?

Risk when camera gets broken.

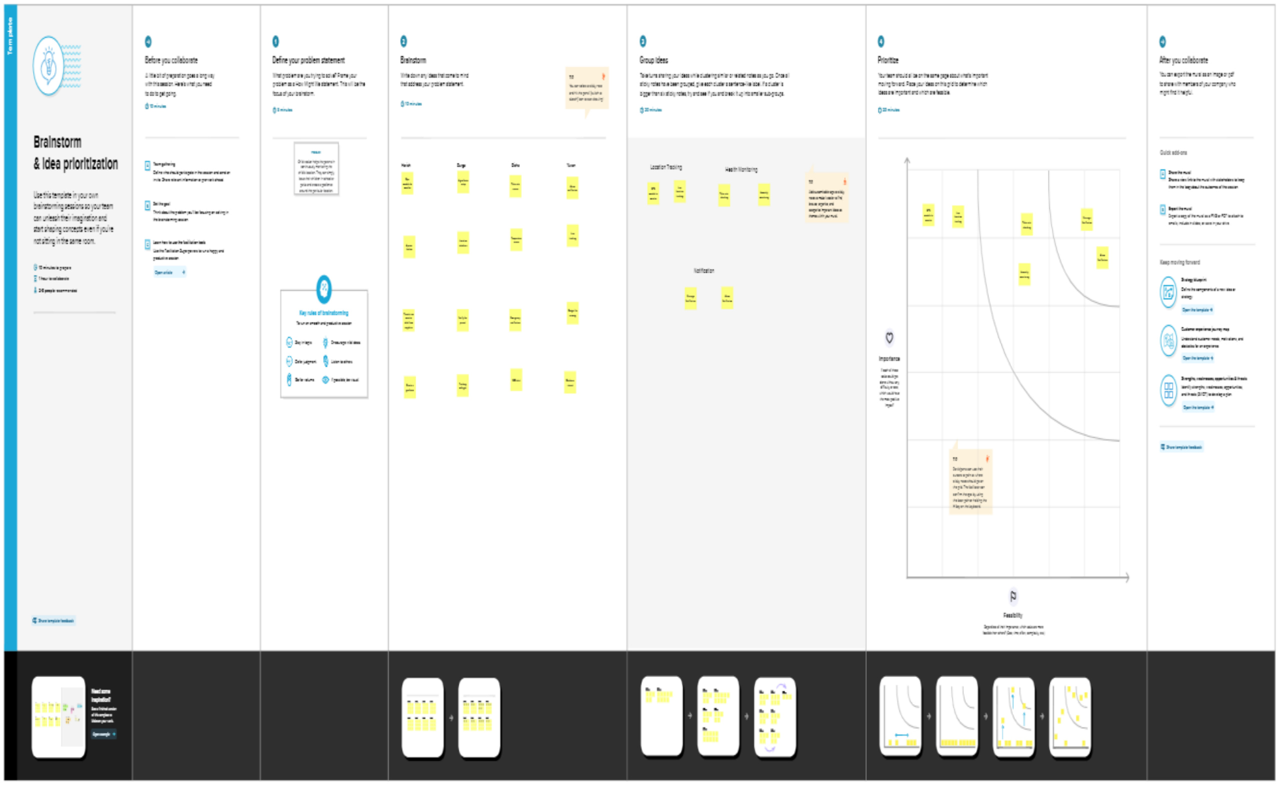
It is not efficient to handle.

GAIN?

Prevent the child from the movement of the baby stroller

Helps to see what does the child does.

3.2 IDEATION & BRAINSTORMING



Brainstorm & Idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Before you collaborate

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

Brainstorm

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

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

Prioritize

After you collaborate

3.3 PROPOSED SOLUTION

Problem Statement

In the majority of homes, both parents will be working. So they are unable to constantly watch over their child's activities. In a crowded setting, there is a chance that the child could get lost and it would be challenging to find them, which will worry the parents.

Idea / Solution description

To design an app that uses the GPS module to display the child's position. The child's location should have a Geo-fence set up around it so that it may be checked on a regular basis to see if the child is inside its boundaries. A notification will automatically be created and delivered to the parents or caregiver if the child steps outside the Geo perimeter. The parents will receive a notification of the child's location once every predetermined period of time.

Novelty / Uniqueness

Different notification methods are provided for various situational circumstances. If a cloud request is made, the device will automatically send a notification to the parents.

Social Impact / Customer Satisfaction

Because the parents will feel more at ease knowing where their children are, and because the kids will have more freedom, a safe and pleasant environment will be created for both the parents and the kids. The ability to track the child's location at any time and from any location puts parents at peace regarding their child's whereabouts.

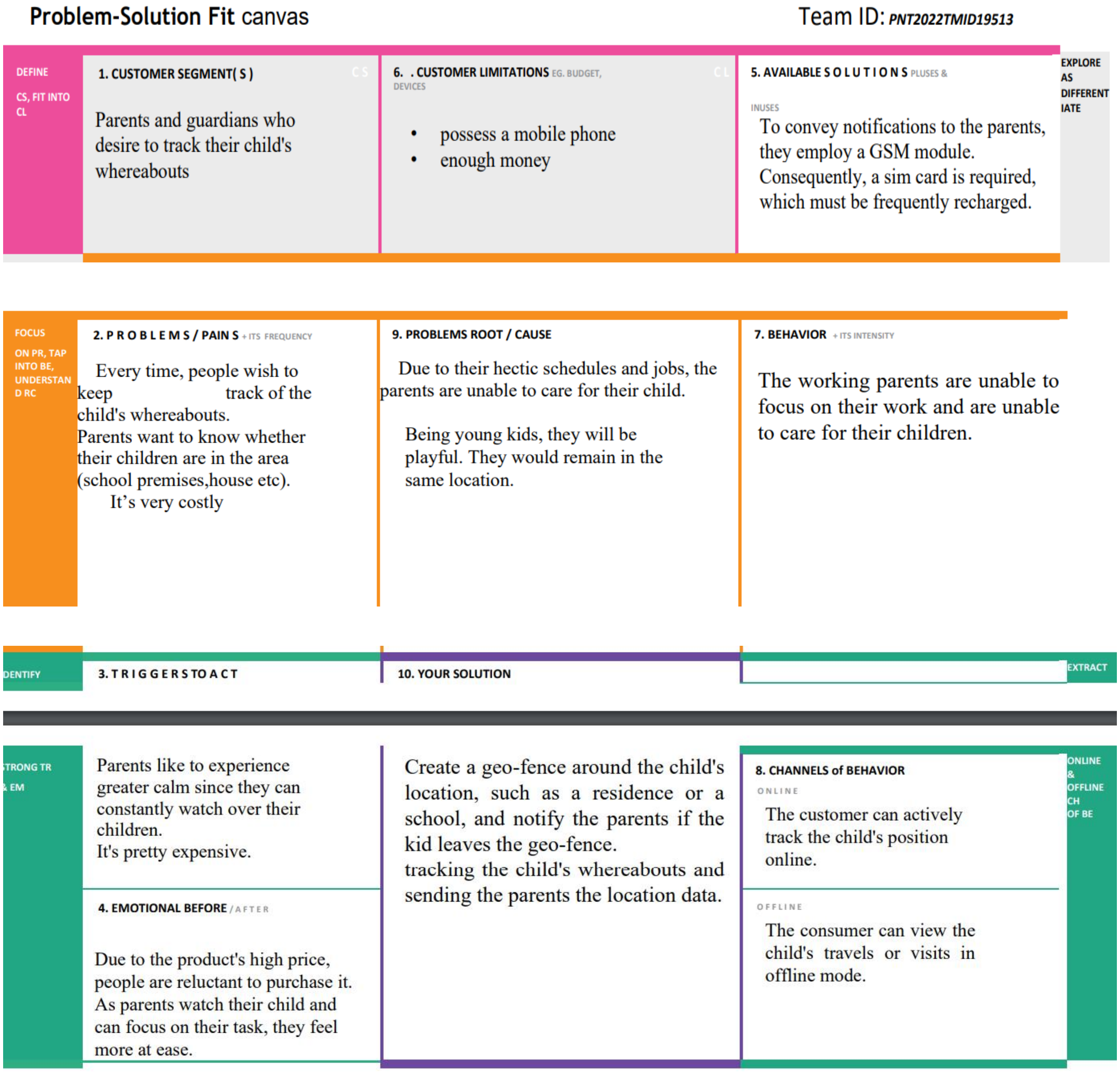
Business Model (Revenue Model)

Providing a gadget and a three-month free subscription to a live tracking and alerting service while selling the product to parents and childcare facilities. If customers enjoy using the device in this way, they can pay to subscribe for a variety of lengths and packages. The parents can feel at ease about their kids while using this technology, allowing them to focus on their work.

Scalability of the Solution

The ongoing monitoring of the child's current location and the recording of the names of previous places visited by the child in a database for use in an emergency.

3.4 PROBLEM SOLUTION FIT



Problem-Solution Fit canvas

Team ID: *PNT2022TMID19513*

DEFINE

CS, FIT INTO CL

1. CUSTOMER SEGMENT(S) CS

Parents and guardians who desire to track their child's whereabouts

6. . CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- possess a mobile phone
- enough money

5. AVAILABLE SOLUTIONS PLUSES & INUSES

To convey notifications to the parents, they employ a GSM module. Consequently, a sim card is required, which must be frequently recharged.

EXPLORE AS DIFFERENTIATE

FOCUS ON PR, TAP INTO BE, UNDERSTAND RC

2. PROBLEMS / PAINS + ITS FREQUENCY

Every time, people wish to keep track of the child's whereabouts.
Parents want to know whether their children are in the area (school premises,house etc).
It's very costly

9. PROBLEMS ROOT / CAUSE

Due to their hectic schedules and jobs, the parents are unable to care for their child.

Being young kids, they will be playful. They would remain in the same location.

7. BEHAVIOR + ITS INTENSITY

The working parents are unable to focus on their work and are unable to care for their children.

DENTIFY STRONG TR & EM

3. TRIGGERS TO ACT

Parents like to experience greater calm since they can constantly watch over their children.
It's pretty expensive.

4. EMOTIONAL BEFORE / AFTER

Due to the product's high price, people are reluctant to purchase it. As parents watch their child and can focus on their task, they feel more at ease.

10. YOUR SOLUTION

Create a geo-fence around the child's location, such as a residence or a school, and notify the parents if the kid leaves the geo-fence.
tracking the child's whereabouts and sending the parents the location data.

EXTRACT ONLINE & OFFLINE CH OF BE

8. CHANNELS of BEHAVIOR

ONLINE

The customer can actively track the child's position online.

OFFLINE

The consumer can view the child's travels or visits in offline mode.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

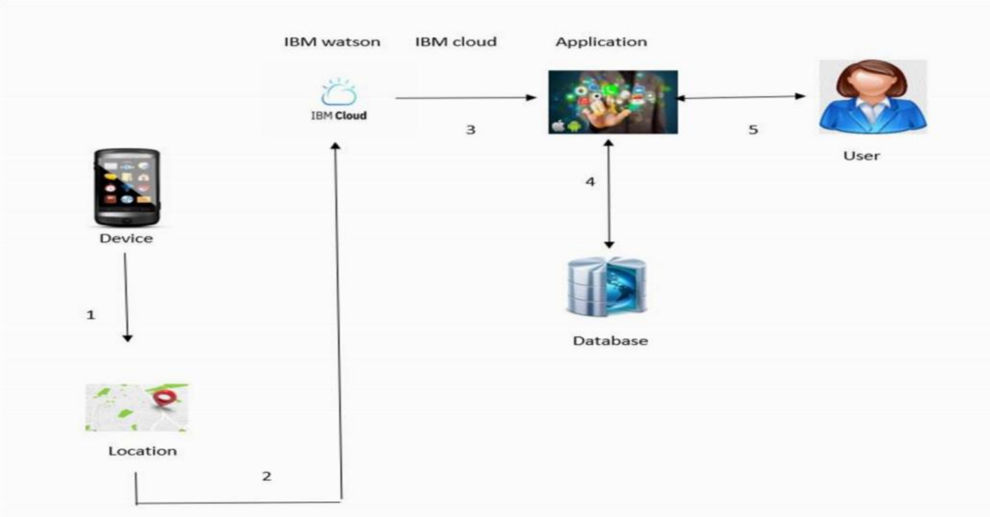
FR No	**Functional Requirements (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through website Registration through app
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Setting up User Id and password
FR-4	App permission	Grant the permission for the app to access location, contact details etc..
FR-5	Interface with the Device	Use the device ID to connect the device with the registered app
FR-6	Setting Geo-Fence	Set the perimeter of area for geo fencing
FR-7	Database	Location history is stored in the cloud Can be accessed from the dashboard
FR-8	Tracking location	Tracking the location through website Tracking the location through app

4.2 NON-FUNCTIONAL REQUIREMENTS

NFR No.	**Non-Functional Requirements**	**Description**
NFR-1	Usability	User-friendly Interface. The device is portable and easy to use. Easily Accessible.
NFR-2	Security	System requires the user to enter a user id and password to access. Location data can only be viewed by the user.
NFR-3	Reliability	If any issues with the gadget are discovered, an update will be given.
NFR-4	Performance	The performance of the device decrease in a network less area. Location tracking will be accurate.
NFR-5	Availability	If there is any update then the device wont be able to operate for an amount of time.
NFR-6	Scalability	Two users can simultaneously watch over the same gadget.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



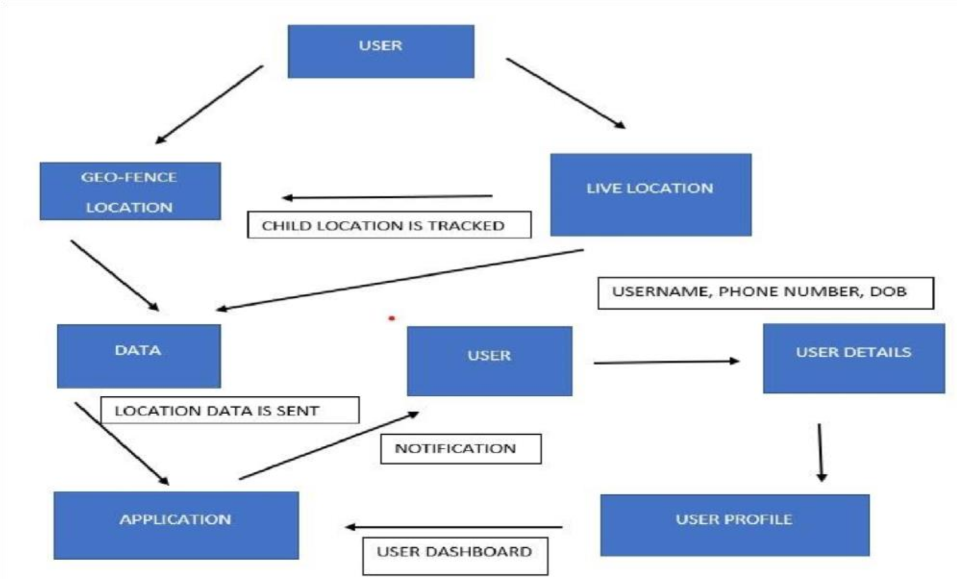
1. GPS tracking device tracks the location of a child.

2. Location data collected from the device is sends to IBM Watson cloud.

3. The Processed data from the IBM Watson cloud is send to User application.

4. The data are stored in the database can be accessed through the application

5. The user can use the application to view the child’s location and the visited location history.



5.2 SOLUTION & TECHNICAL ARCHITECTURE

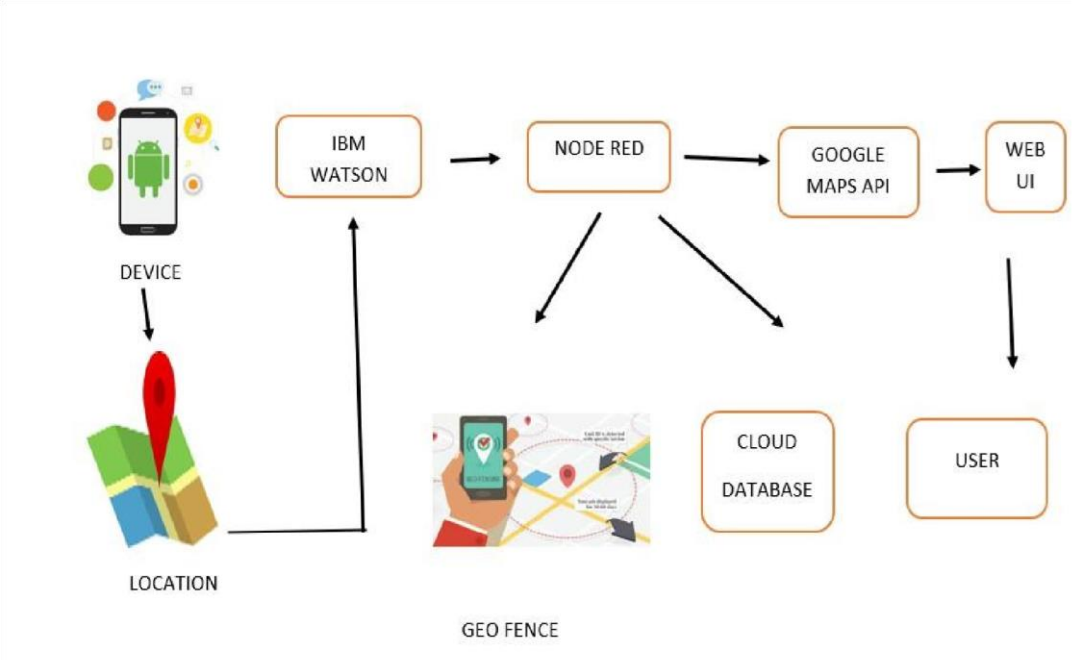


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with the device through the mobile application.	MIT app inventor.
2.	Application Logic-1	The data from the device are collected and sent to the application	IBM Watson

3.	Application Logic-2	Connect the device to the internet and the application	Node RED
4.	Database	Child’s visited location history will be stored	MySQL
5.	Cloud Database	Database Service on Cloud	IBM Cloudant
6.	File Storage	File’s based on the location are stored	IBM Block Storage or Other Storage Service
7.	External API-1	To create the geo fence and monitor the location	Google maps API
8.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Cloud Foundry

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	List the open-source frameworks used	MIT app inventor
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption
3.	Scalable Architecture	Update for the application is provided every time if an error is found	cloud

4.	Availability	The application has to be available every time without any problems	IBM cloud
5.	Performance	The device works best with good network	High speed and stable network

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application 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 for the application	I can receive confirmation email & click confirm	High	Sprint-1

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can log in to the application with the email & password	High	Sprint-1
	Dashboard	USN-5	As a user I can log into the dashboard and use the functions in it	I can view the location of the child and see the visited location history	High	Sprint-4
Customer Care Executive	Help	USN-6	As a user I can contact the administrator through Help option	I can contact administrator in case of issues in app/devices	Medium	Sprint-4
Administrator	Administration	USN-7	As an administrator I	I can provide	High	Sprint-3

			can ensure the safety of the data and error free application	the apps if there is any problems in the application/device		

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-1	As a user, I can login to the webpage	2	High	Harish, Durga
Sprint-2	Login	USN-2	Creating a MIT APP with login page	2	High	Disha, Yuvan
Sprint-3	Creating a Geo-fence	USN-3	As a user I can create a geo-fence around a location that is required	2	Medium	Durga, Yuvan
Sprint-4	Notification	USN-4	As a user I can receive notification for certain events	2	High	Harish, Durga, Yuvan

6.2 SPRINT DELIVERY SCHEDULE

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

6.3 REPORTS FROM JIRA

https://www.atlassian.com/agile/project-management

http://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

http://www.atlassian.com/agile/tutorials/epics

http://www.atlassian.com/agile/tutorials/sprints

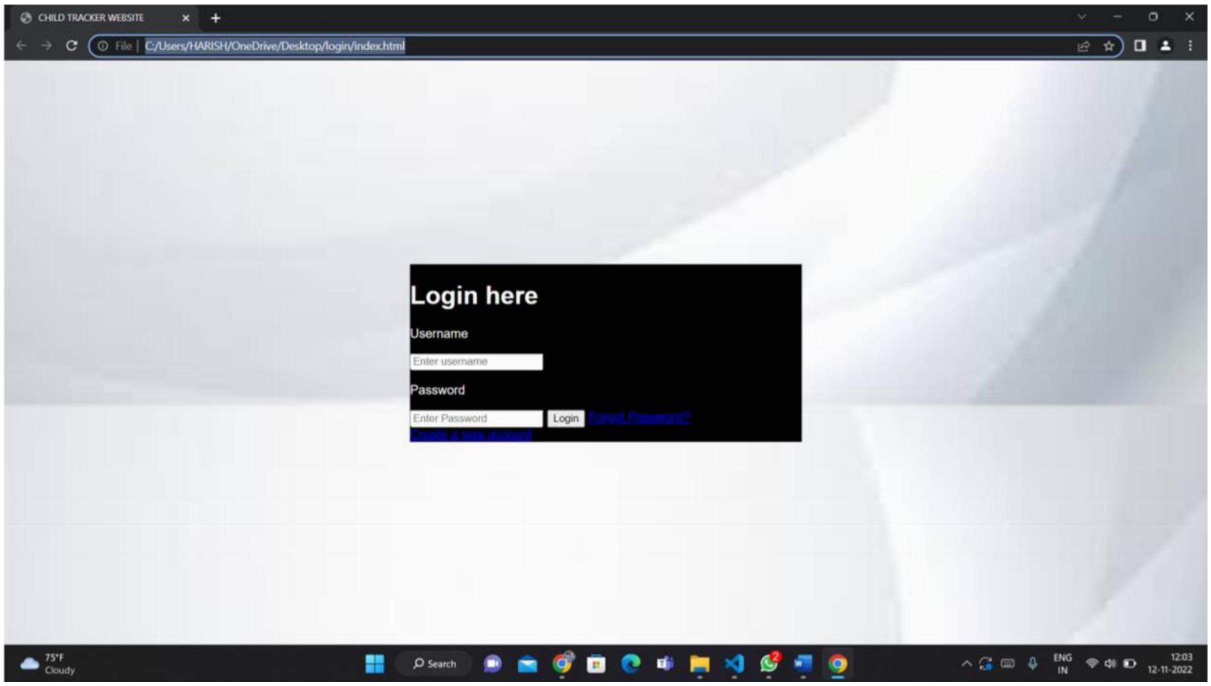
http://www.atlassian.com/agile/project-management/estimation

http://www.atlassian.com/agile/tutorials/burndown-charts

7.CODING AND SOLUTIONING

7.1 SPRINT – 1

SPRINT1: Create a Login Page



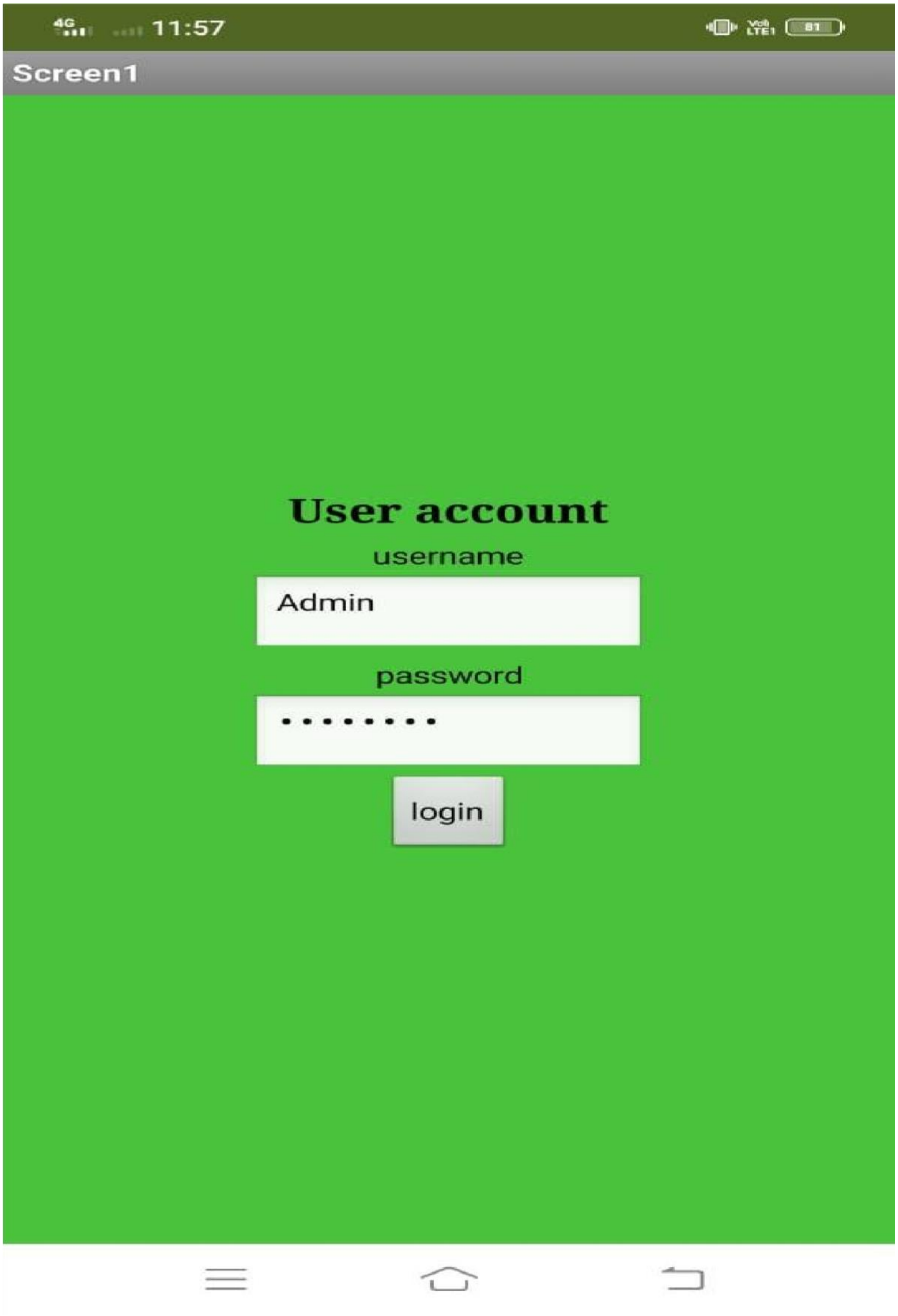
HTML CODE:

```
<!DOCTYPE html>
<head>
    <title>
        CHILD TRACKER WEBSITE
    </title>
    <link rel="stylesheet" type="text/css" href="style.css">

    <body>

<div class="loginbox">

    <h1>

        Login here

    </h1>
    <form>

        <p>

            Username

        </p>

        <input type="text" name="" placeholder="Enter username">

        <p> Password</p>

        <input type="password" name="" placeholder="Enter Password">

        <input type="submit" name="" value="Login">

        <a href="#">Forgot Password?</a><br>

        <a href="#">Create a new account</a>


    </form>

</div>

    </body>

</head></html>
```

```
CSS CODE:  body{

margin: 0%;     padding: 0%;
```

```
    background:url(background.jpg);     background-size:
cover;     background-position: center;     font-family:
sans-serif;


}
.loginbox{     width:500px;     heigth:500px;
background: #000;     color:#fff;     top:50%;
left:50%;     position: absolute;     transform:
translate(-50%,-50%);     box-sizing:border-box'

}
```

7.2 SPRINT 2



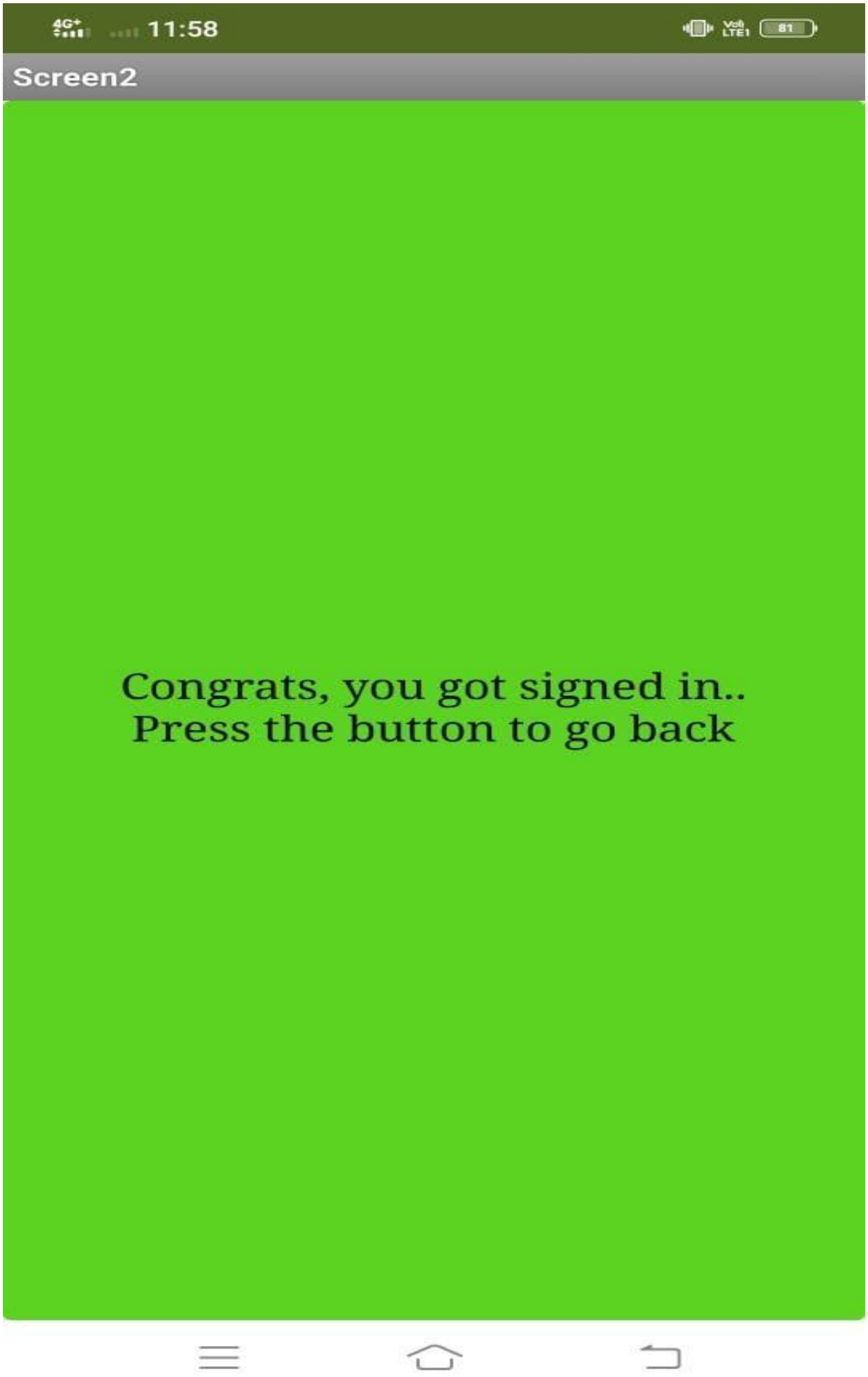
User account

username

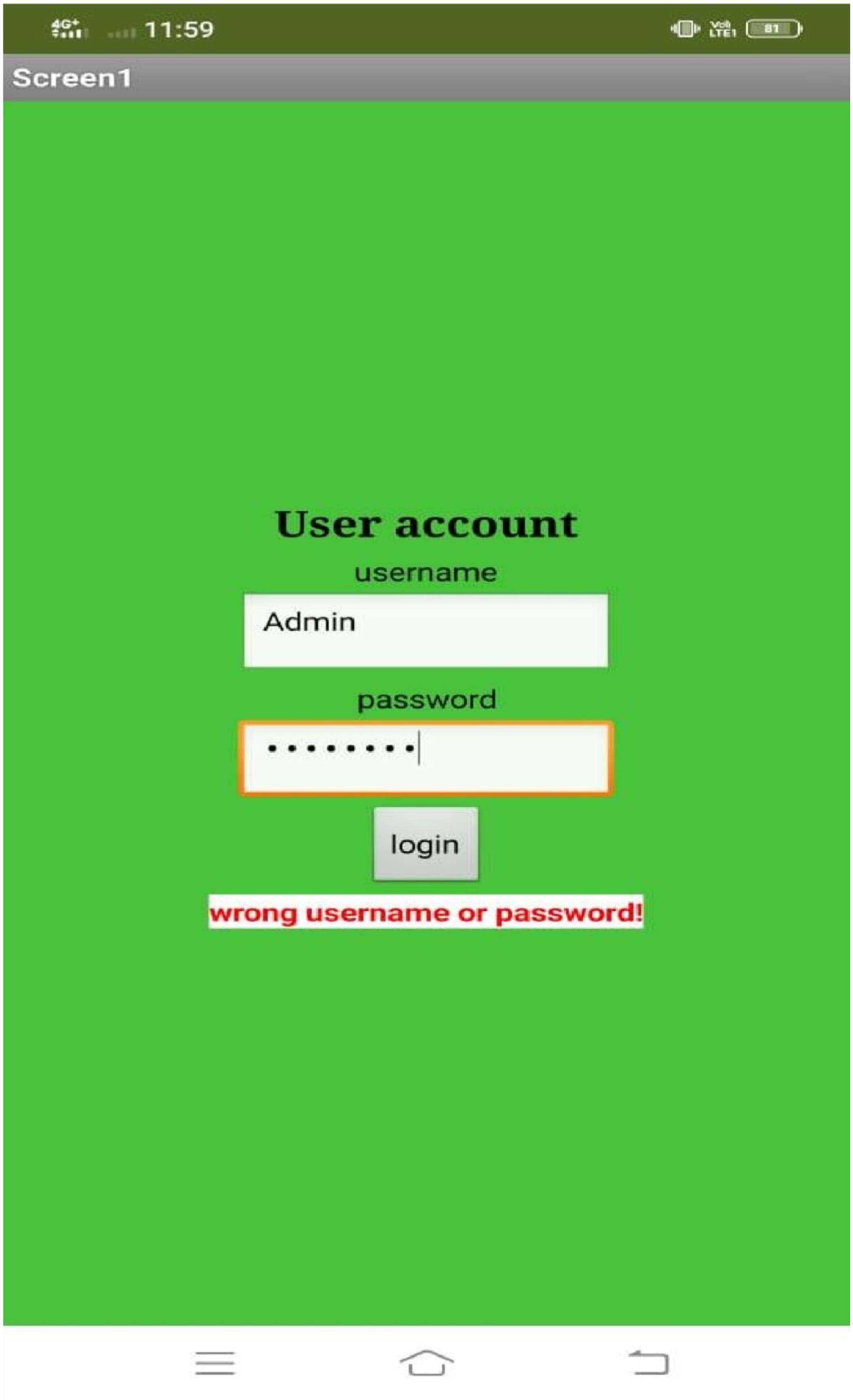
Admin

password

•••••••••



Congrats, you got signed in..
Press the button to go back



User account

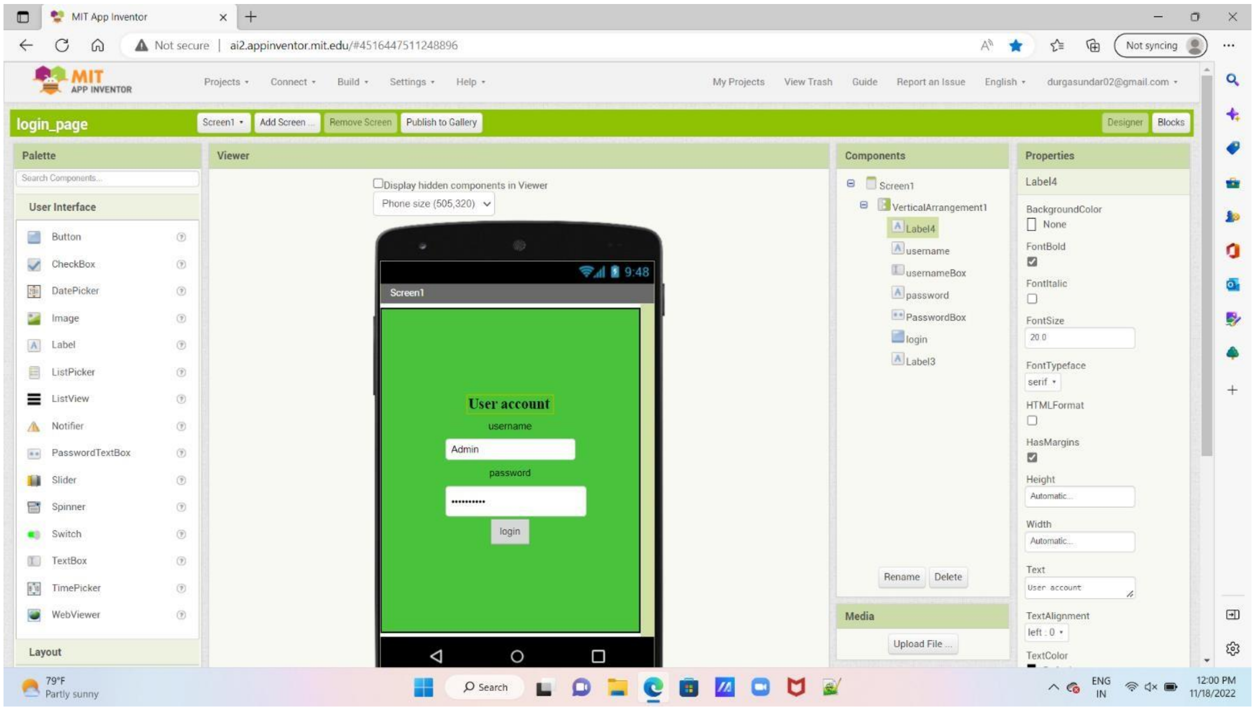
username

Admin

password

•••••••••

wrong username or password!



Label4

BackgroundColor
□ None

FontBold
☑

FontItalic
□

FontSize
20.0

FontTypeface
serif

HTMLFormat
□

HasMargins
☑

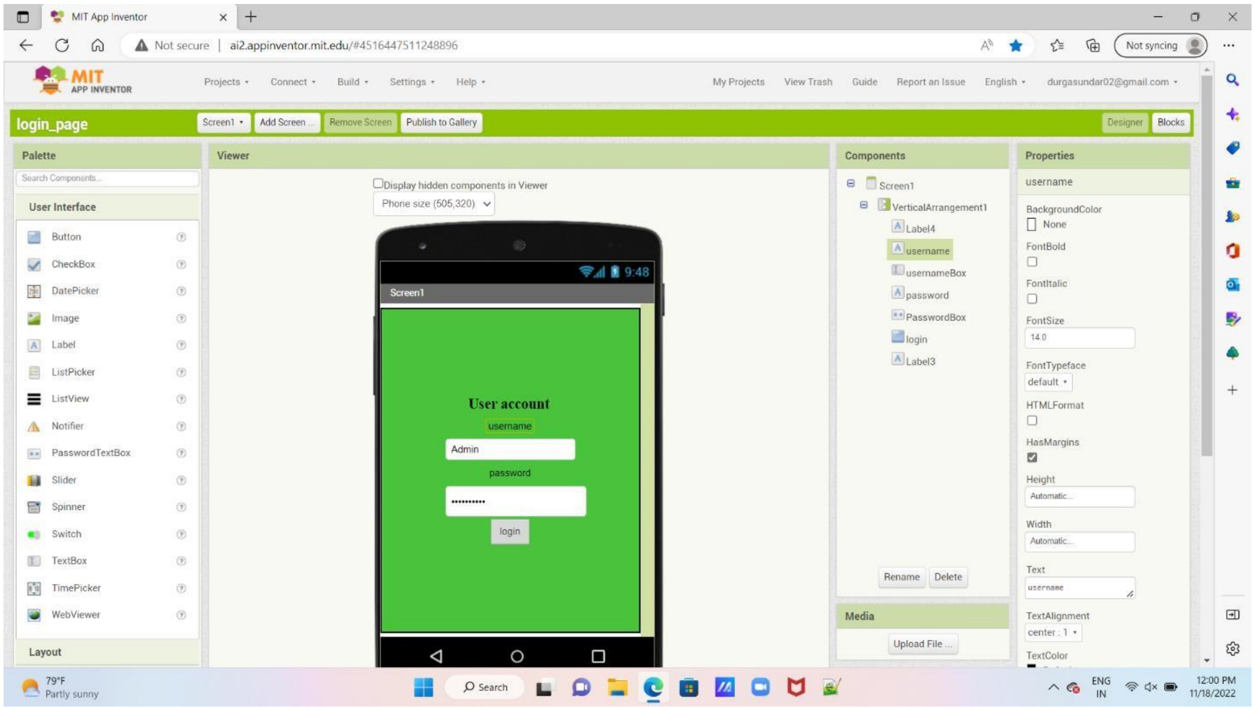
Height
Automatic...

Width
Automatic...

Text
User account

TextAlignment
left : 0

TextColor



username

BackgroundColor
□ None

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default

HTMLFormat
□

HasMargins
☑

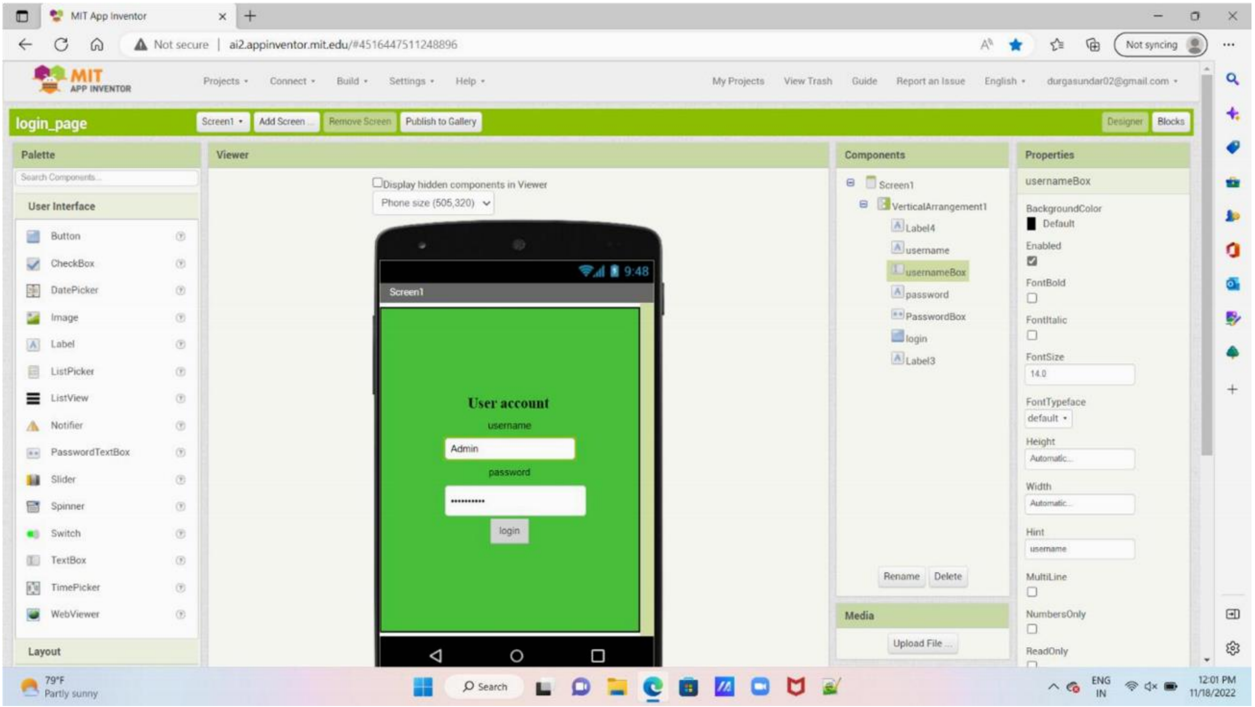
Height
Automatic...

Width
Automatic...

Text
username

TextAlignment
center : 1

TextColor



Components

- Screen1
  - VerticalArrangement1
    - Label4
    - username
    - usernameBox
    - password
    - PasswordBox
    - login
    - Label3

Properties

usernameBox

BackgroundColor
■ Default

Enabled
☑

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default ▾

Height
Automatic...

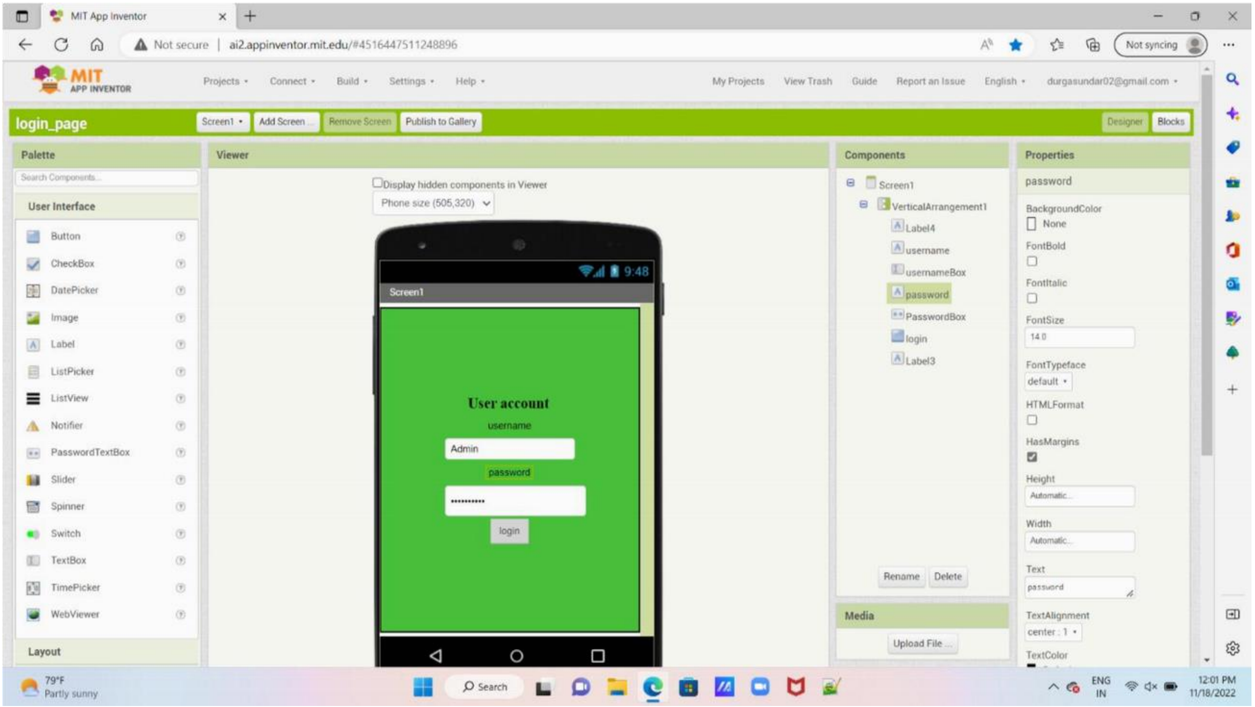
Width
Automatic...

Hint
username

MultiLine
□

NumbersOnly
□

ReadOnly
□



Components

- Screen1
  - VerticalArrangement1
    - Label4
    - username
    - usernameBox
    - password
    - PasswordBox
    - login
    - Label3

Properties

password

BackgroundColor
□ None

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default ▾

HTMLFormat
□

HasMargins
☑

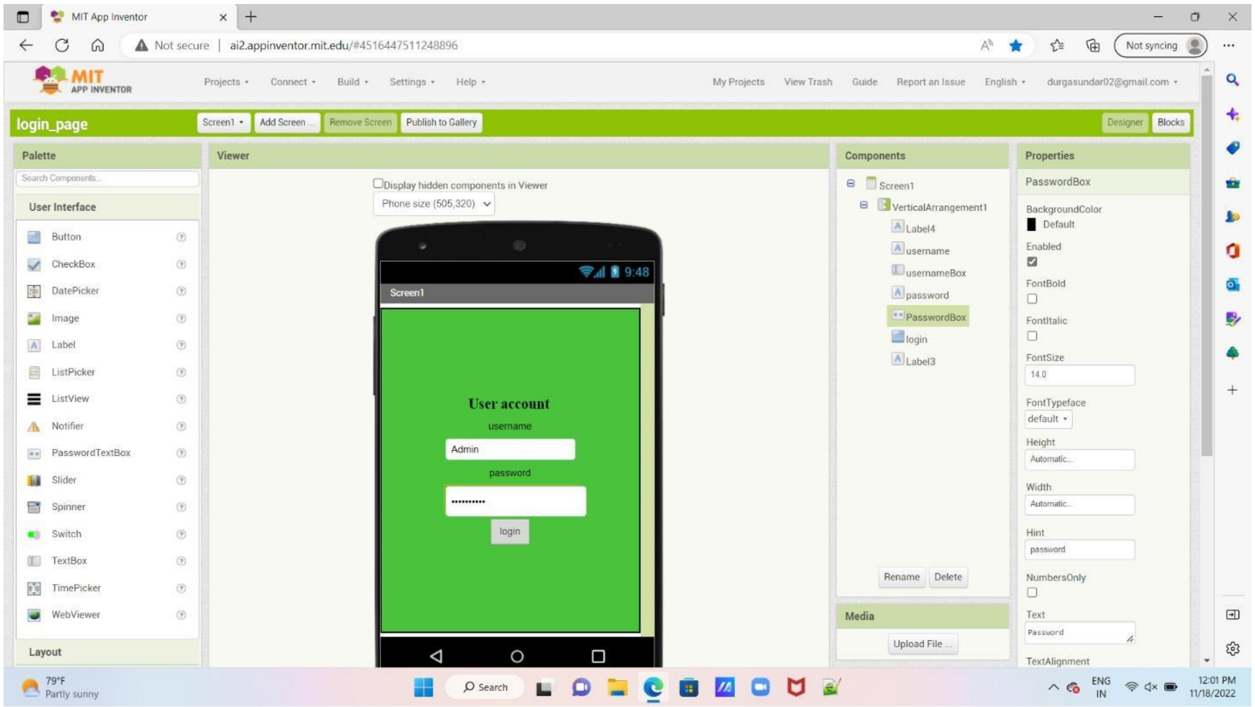
Height
Automatic...

Width
Automatic...

Text
password

TextAlignment
center : 1 ▾

TextColor



User account

username

Admin

password

••••••••••

login

- Screen1
  - VerticalArrangement1
    - Label4
    - username
    - usernameBox
    - password
    - PasswordBox
    - login
    - Label3

PasswordBox

BackgroundColor: ■ Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default ▾

Height: Automatic...

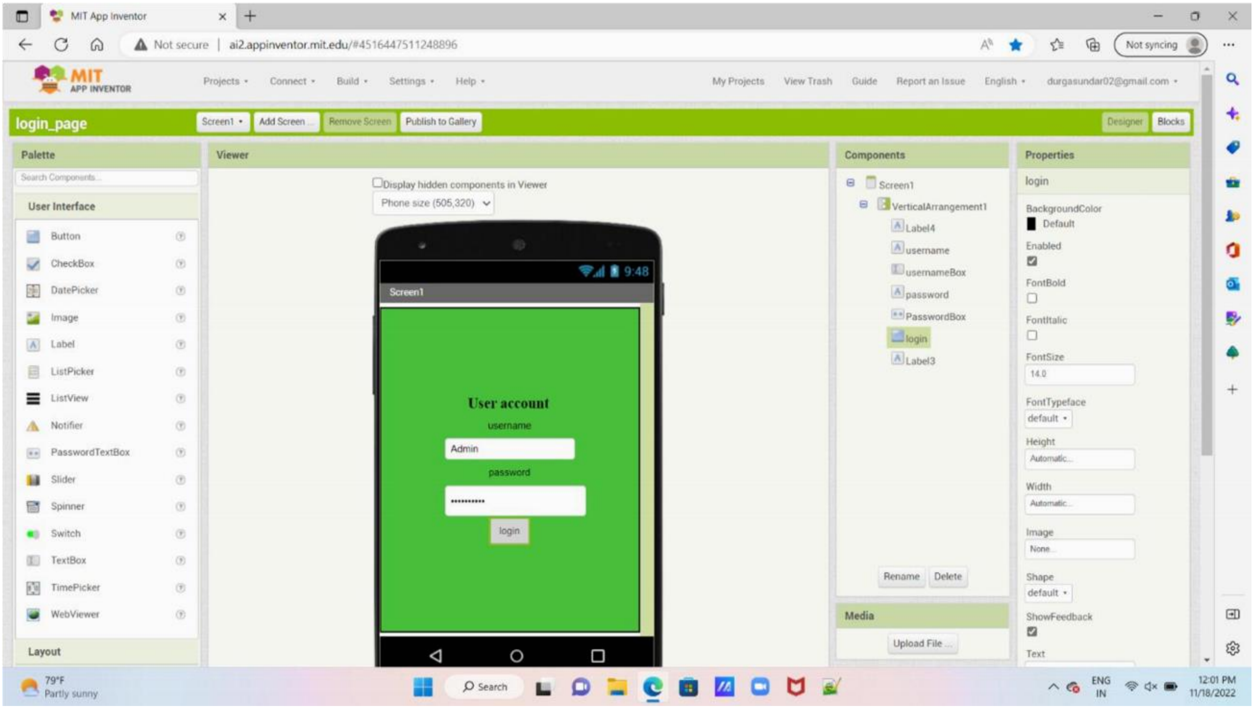
Width: Automatic...

Hint: password

NumbersOnly: □

Text: Password

TextAlignment



User account

username

Admin

password

••••••••••

login

- Screen1
  - VerticalArrangement1
    - Label4
    - username
    - usernameBox
    - password
    - PasswordBox
    - login
    - Label3

login

BackgroundColor: ■ Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default ▾

Height: Automatic...

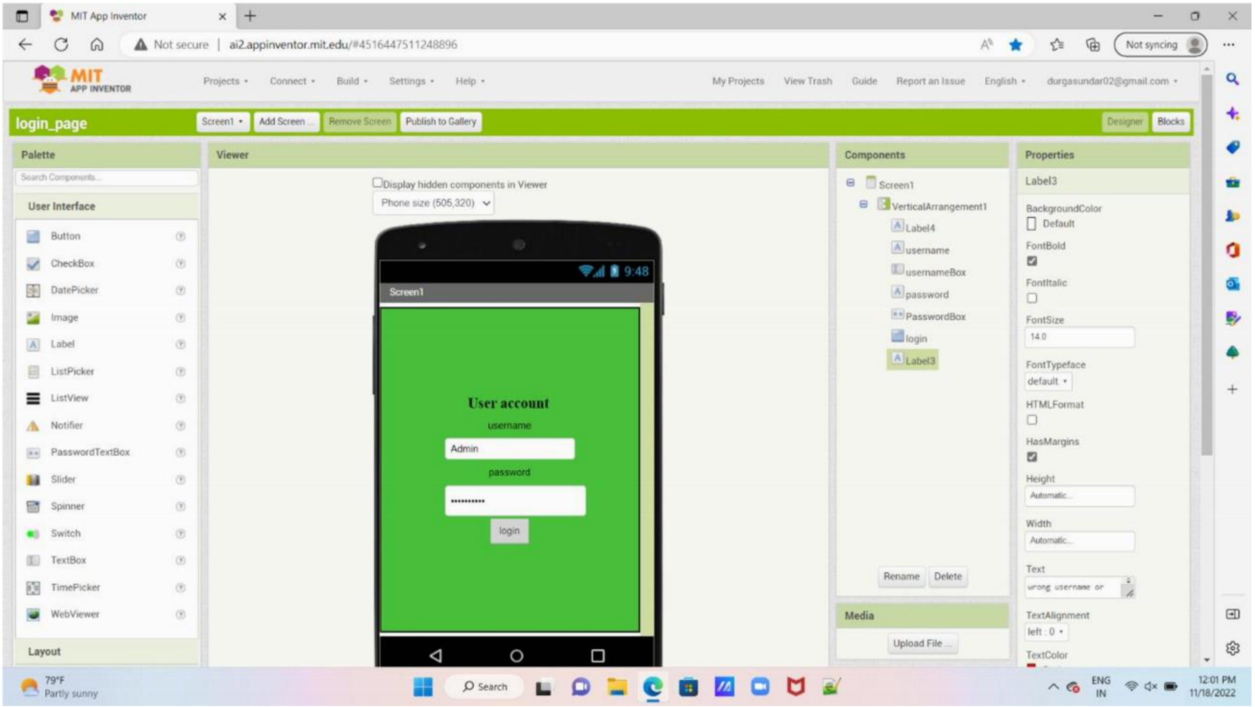
Width: Automatic...

Image: None...

Shape: default ▾

ShowFeedback: ☑

Text



Label3

BackgroundColor
□ Default

FontBold
☑

FontItalic
□

FontSize
14.0

FontTypeface
default

HTMLFormat
□

HasMargins
☑

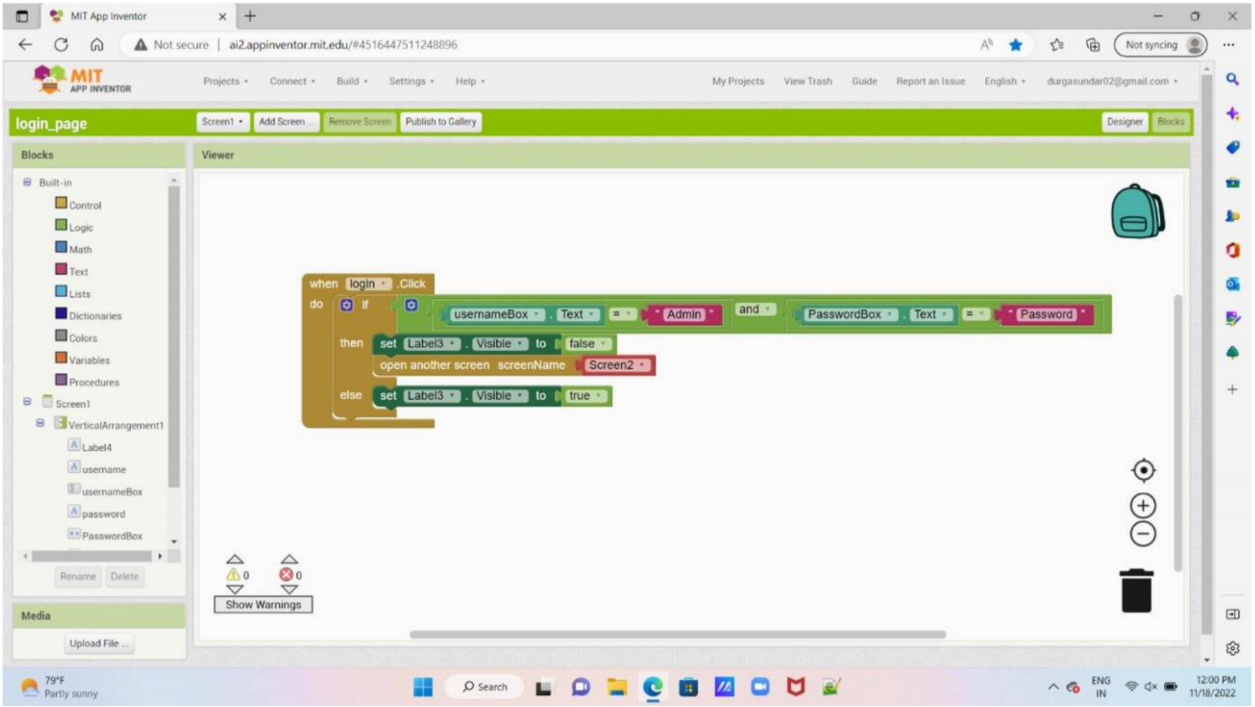
Height
Automatic...

Width
Automatic...

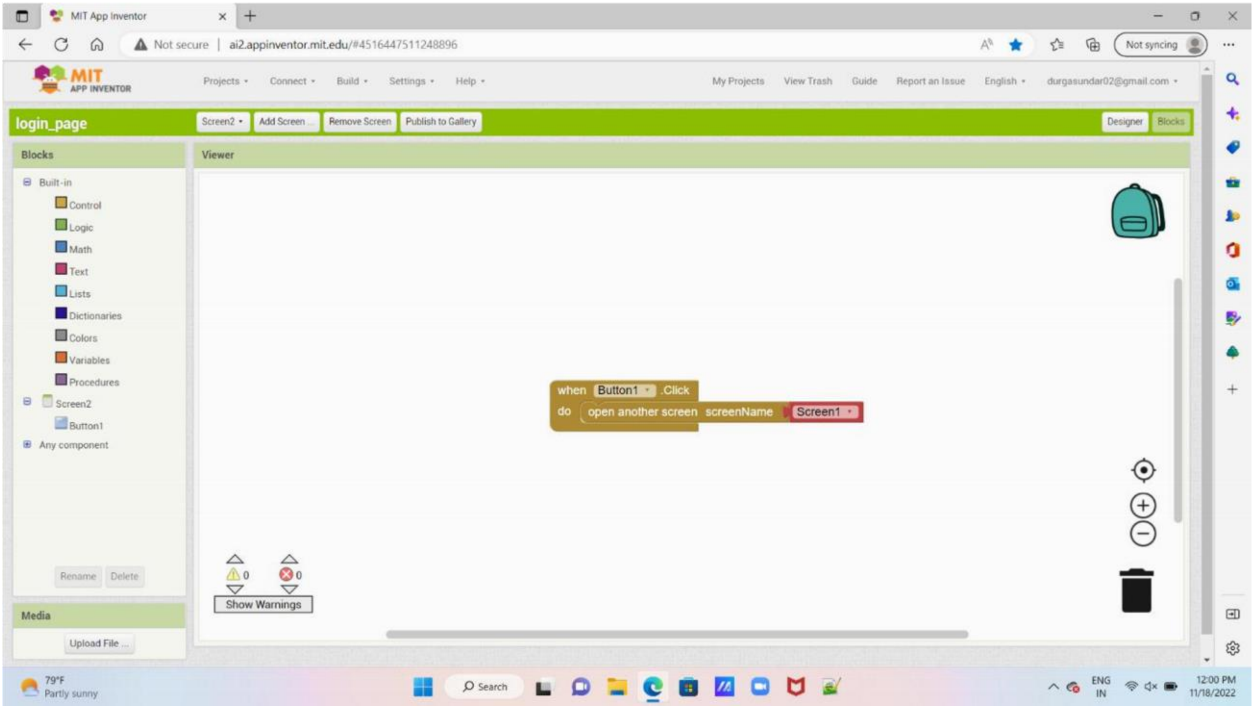
Text
wrong username or

TextAlignment
left : 0

TextColor



```
when login .Click
do  if    usernameBox . Text = "Admin" and PasswordBox . Text = "Password"
    then  set Label3 . Visible to false
          open another screen screenName Screen2
    else  set Label3 . Visible to true
```



```
when Button1 .Click
do  open another screen  screenName  Screen1
```

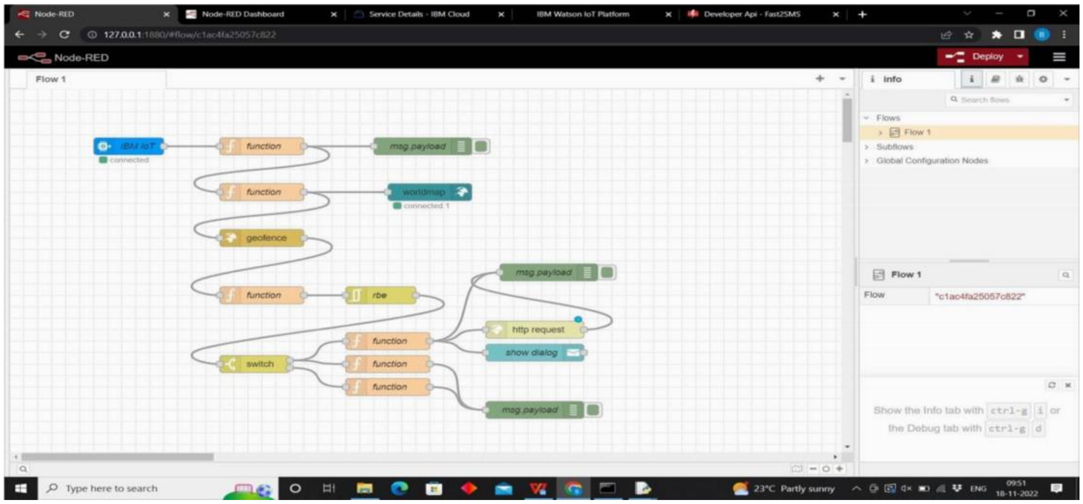
7.3 SPRINT 3

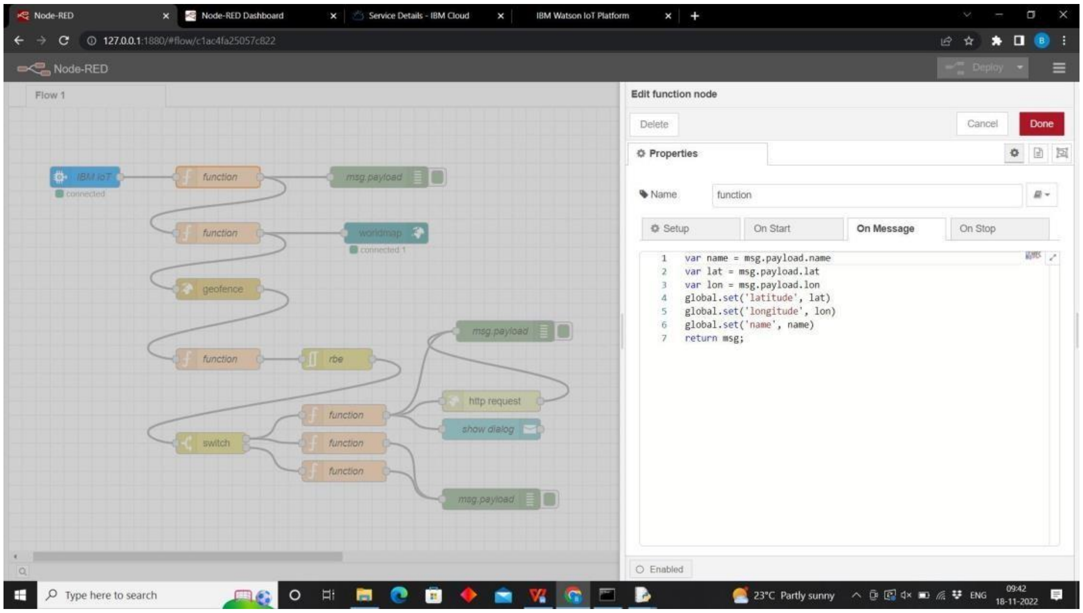
Python program :

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


    myConfig={

"identity":{

          "orgId":"hi70w8",

          "typeId":"gps",

          "deviceId":"987654321"

},

"auth":{

          "token":"24688462"

}

    }

    client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=Non e)
    client.connect()


    while True: name="GPS"


#outside

#latitude=10.820155

#longitude=77.016172


#inside latitude=10.826579
longitude=77.059943
myData={'name':name,'lat':latitude,'lon':longit
ude}
```

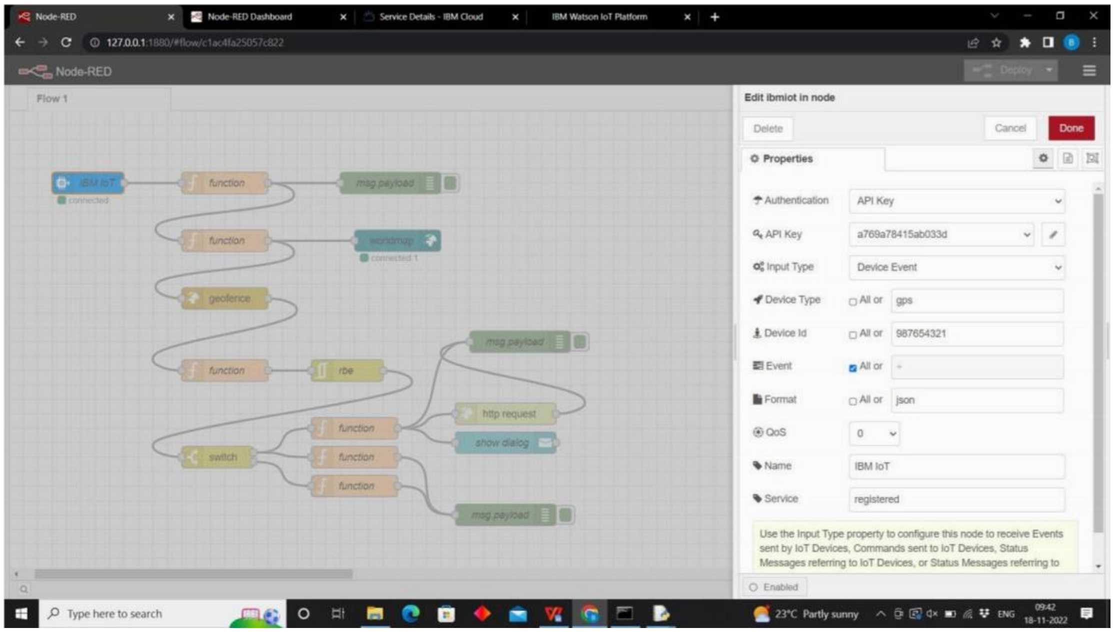
```
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
print("Data published to IBM platform:",myData) time.sleep(5)
    client.disconnect()
```

Node Red:





```
var name = msg.payload.name
var lat = msg.payload.lat
var lon = msg.payload.lon
global.set('latitude', lat)
global.set('longitude', lon)
global.set('name', name)
return msg;
```



Edit ibmiot in node

Delete Cancel Done

Properties

Authentication: API Key

API Key: a769a78415ab033d

Input Type: Device Event

Device Type: □ All or gps

Device Id: □ All or 987654321

Event: ☑ All or -

Format: □ All or json

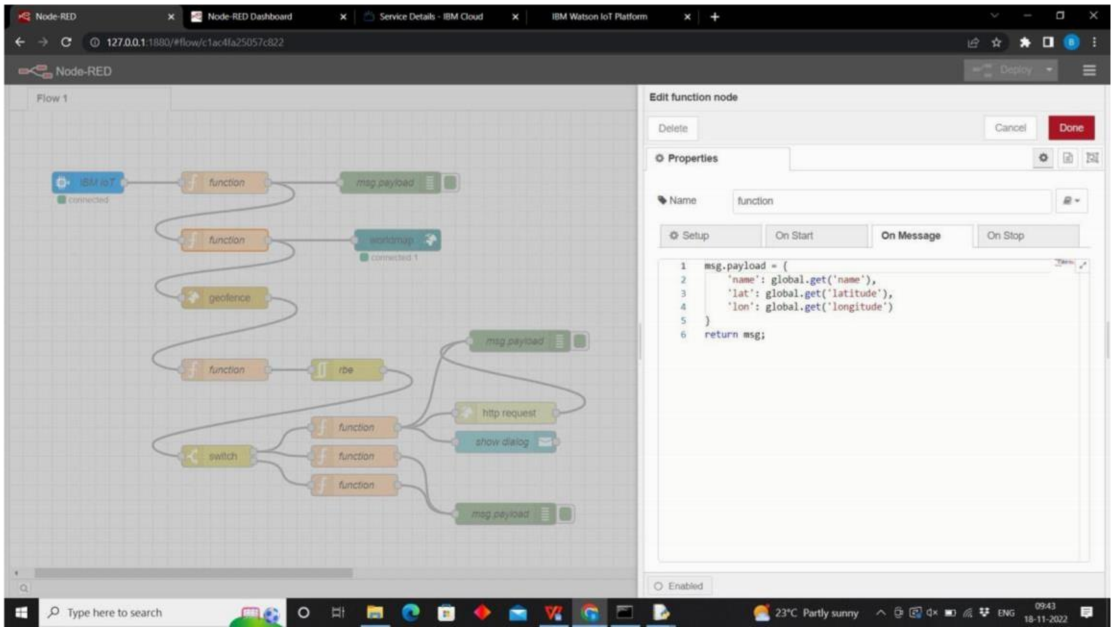
QoS: 0

Name: IBM IoT

Service: registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to

○ Enabled



Edit function node

Delete Cancel Done

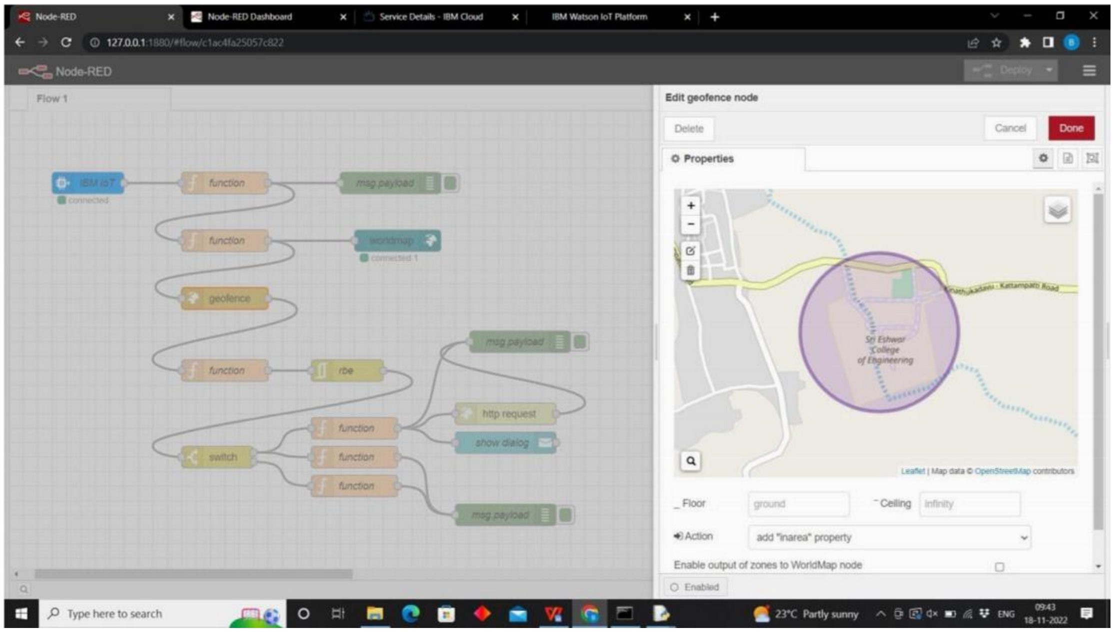
Properties

Name: function

Setup | On Start | On Message | On Stop

```
msg.payload = {
    'name': global.get('name'),
    'lat': global.get('latitude'),
    'lon': global.get('longitude')
}
return msg;
```

○ Enabled



Edit geofence node

Delete Cancel Done

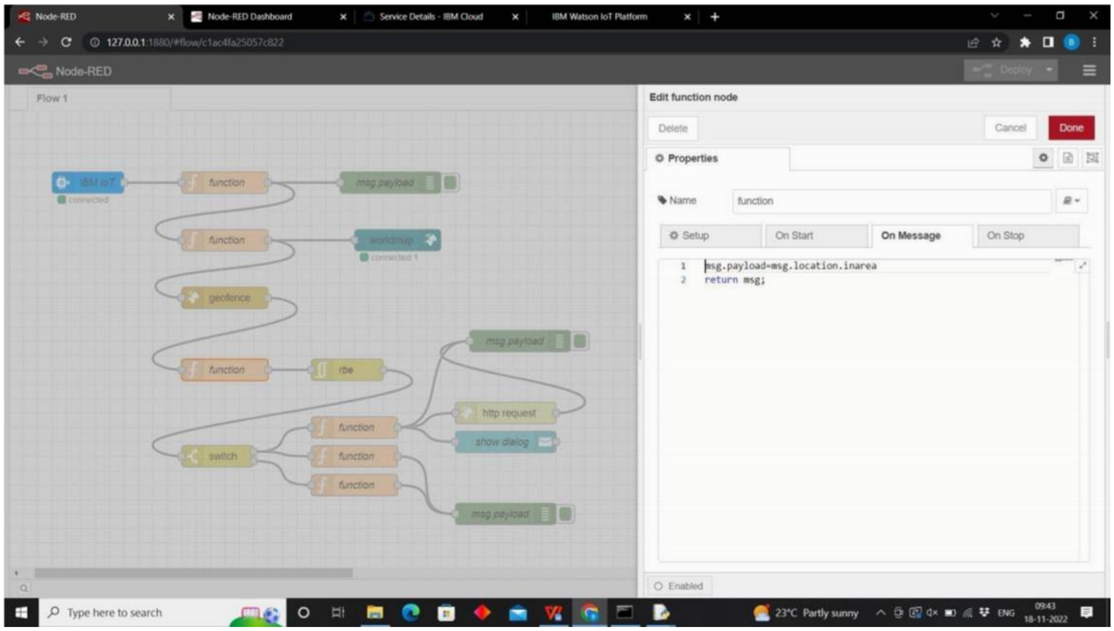
Properties

_ Floor: ground   ‾ Ceiling: infinity

Action: add "inarea" property

Enable output of zones to WorldMap node □

○ Enabled



Edit function node

Delete Cancel Done

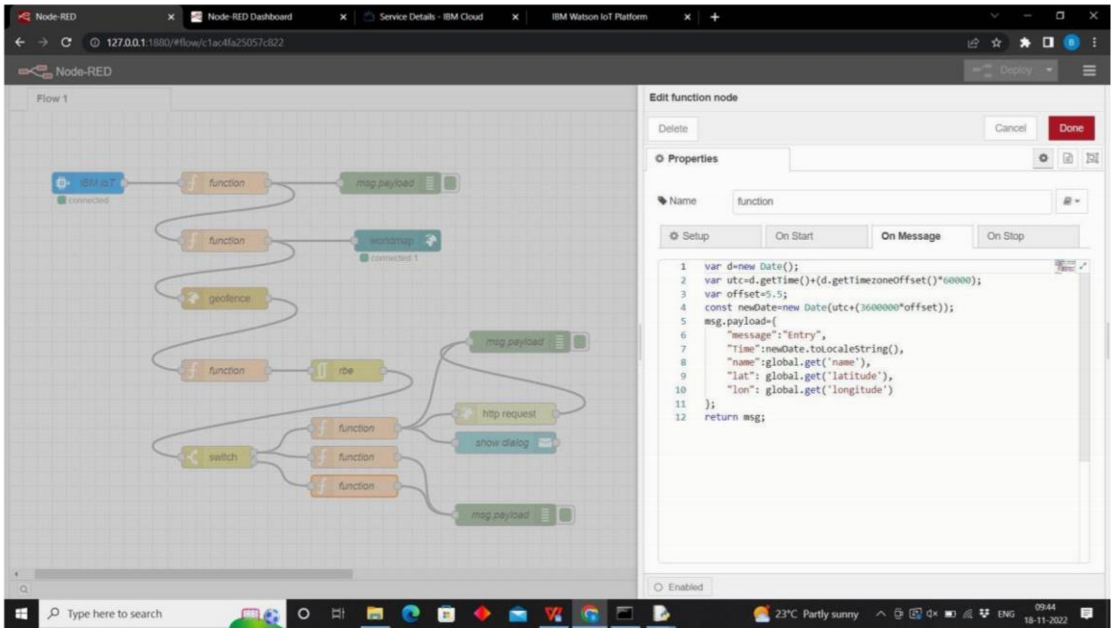
Properties

Name: function

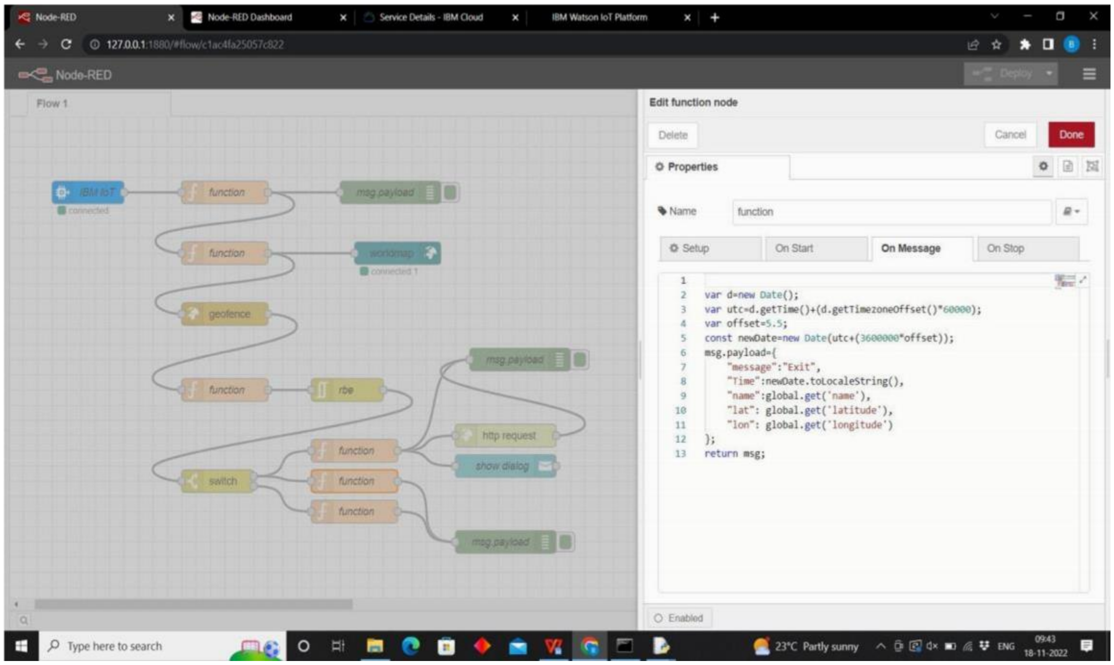
Setup | On Start | On Message | On Stop

```
msg.payload=msg.location.inarea
return msg;
```

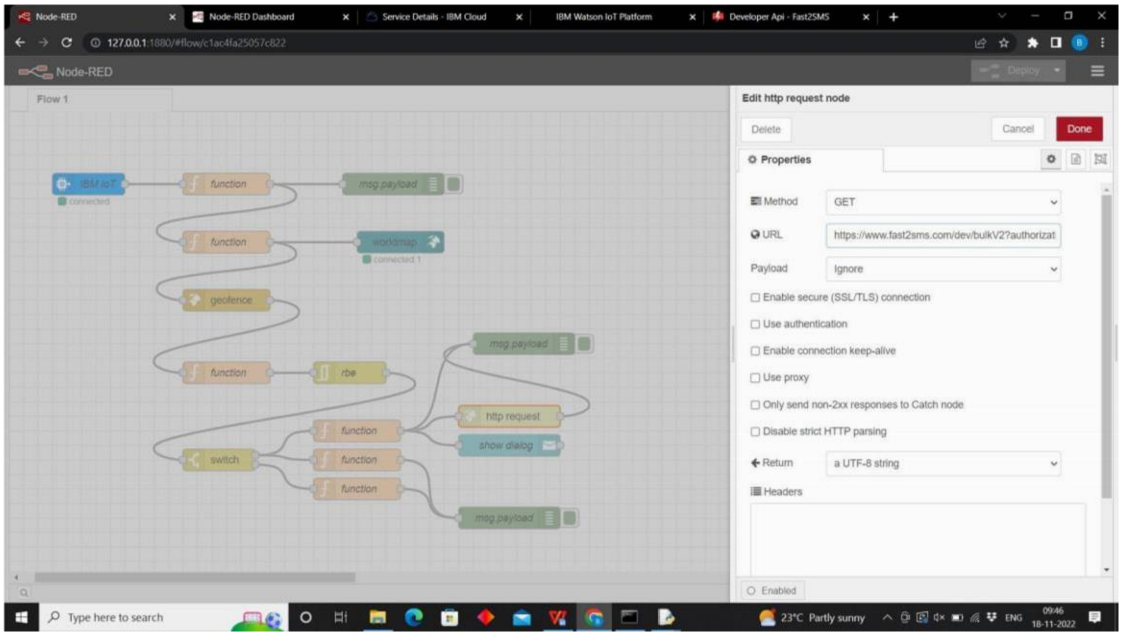
○ Enabled



```
var d=new Date();
var utc=d.getTime()+(d.getTimezoneOffset()*60000);
var offset=5.5;
const newDate=new Date(utc+(3600000*offset));
msg.payload={
    "message":"Entry",
    "Time":newDate.toLocaleString(),
    "name":global.get('name'),
    "lat": global.get('latitude'),
    "lon": global.get('longitude')
};
return msg;
```



```

var d=new Date();
var utc=d.getTime()+(d.getTimezoneOffset()*60000);
var offset=5.5;
const newDate=new Date(utc+(3600000*offset));
msg.payload={
    "message":"Exit",
    "Time":newDate.toLocaleString(),
    "name":global.get('name'),
    "lat": global.get('latitude'),
    "lon": global.get('longitude')
};
return msg;
```



Edit http request node

Delete Cancel Done

Properties

Method: GET

URL: https://www.fast2sms.com/dev/bulkV2?authorizat

Payload: Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

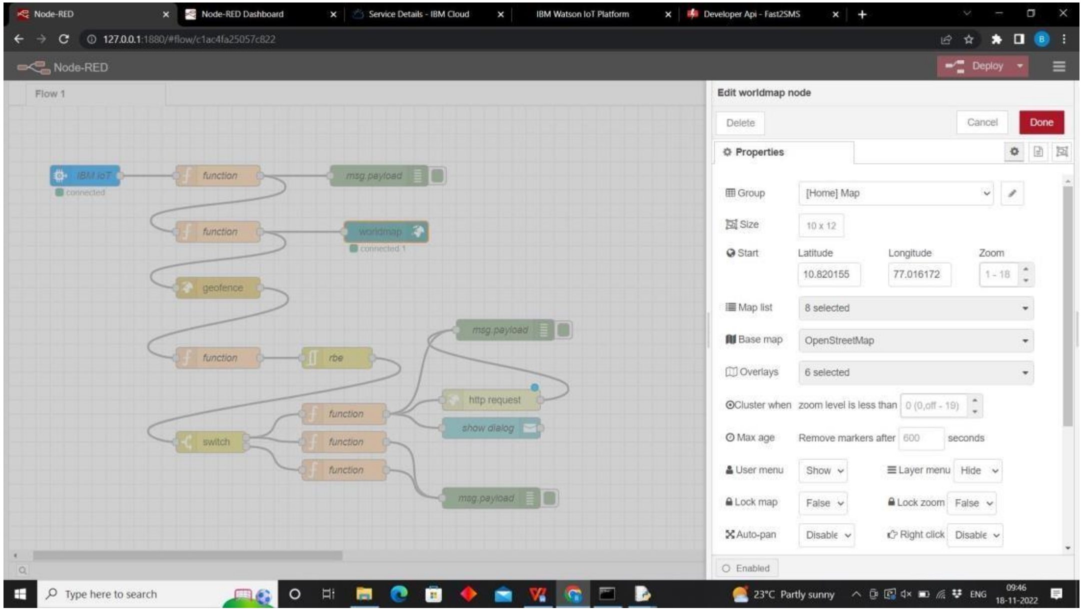
☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

Return: a UTF-8 string

Headers

○ Enabled



Edit worldmap node

Delete Cancel Done

Properties

Group: [Home] Map

Size: 10 x 12

Start: Latitude 10.820155 Longitude 77.016172 Zoom 1 - 18

Map list: 8 selected

Base map: OpenStreetMap

Overlays: 6 selected

Cluster when zoom level is less than 0 (0,off - 19)

Max age: Remove markers after 600 seconds

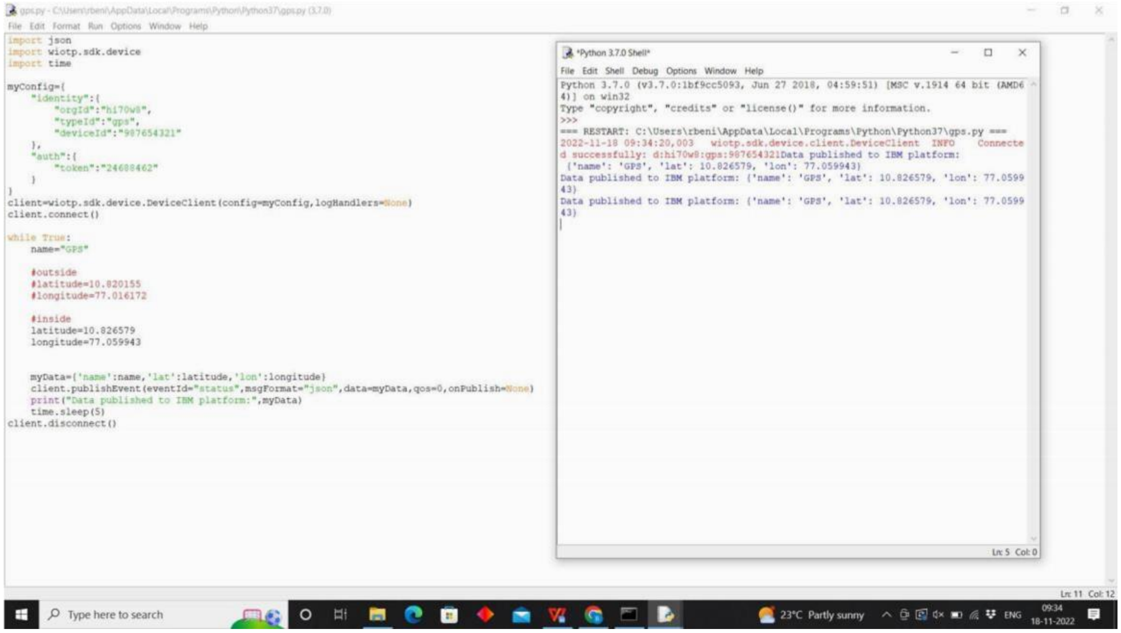
User menu: Show Layer menu: Hide

Lock map: False Lock zoom: False

Auto-pan: Disable Right click: Disable

○ Enabled

Output:



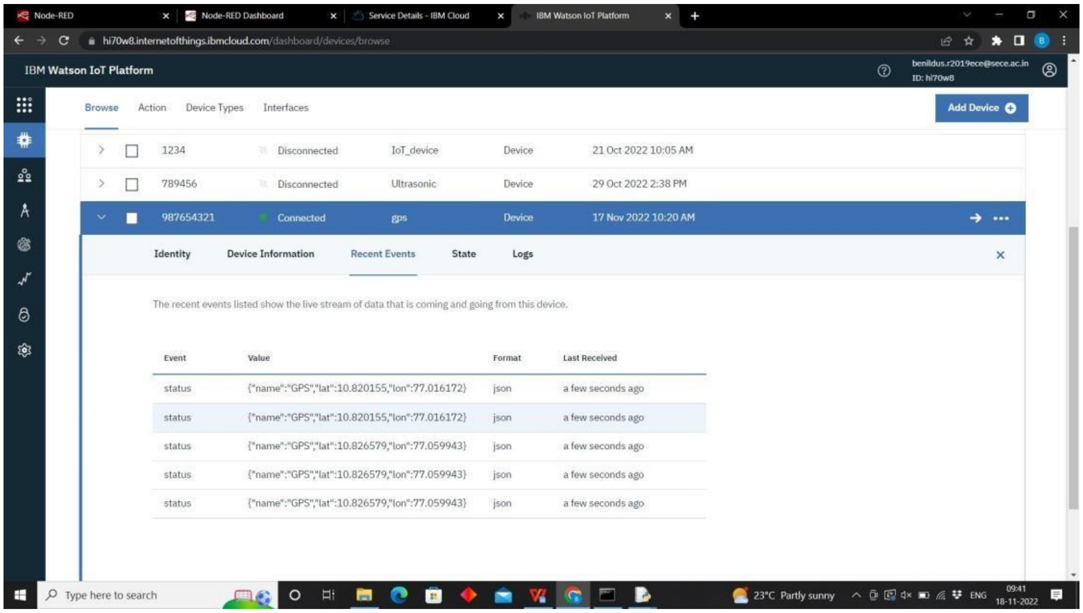
gps.py - C:\Users\rbeni\AppData\Local\Programs\Python\Python37\gps.py (3.7.0)

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

myConfig={
    "identity":{
        "orgId":"hi70w8",
        "typeId":"gps",
        "deviceId":"987654321"
    },
    "auth":{
        "token":"24688462"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()

while True:
    name="GPS"

    #outside
    #latitude=10.820155
    #longitude=77.016172

    #inside
    latitude=10.826579
    longitude=77.059943


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

Python 3.7.0 Shell

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\rbeni\AppData\Local\Programs\Python\Python37\gps.py ===
2022-11-18 09:34:20,003   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:hi70w8:gps:987654321Data published to IBM platform:
 {'name': 'GPS', 'lat': 10.826579, 'lon': 77.059943}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
```

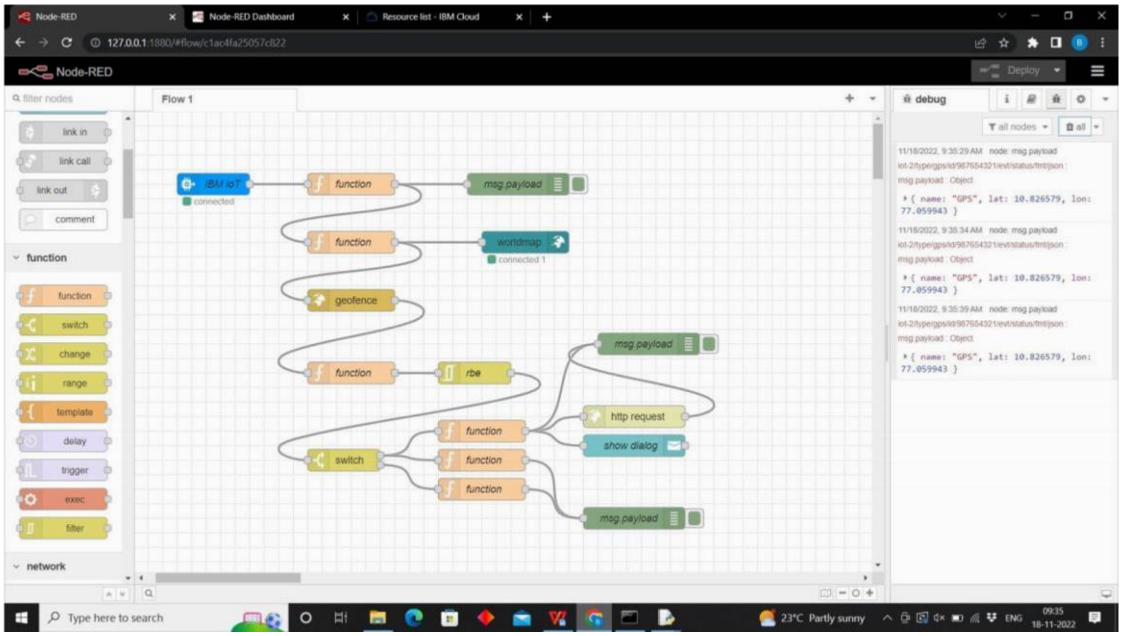
File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help
Ln: 5 Col: 0
Ln: 11 Col: 12
Type here to search
23°C Partly sunny
ENG
09:34
18-11-2022



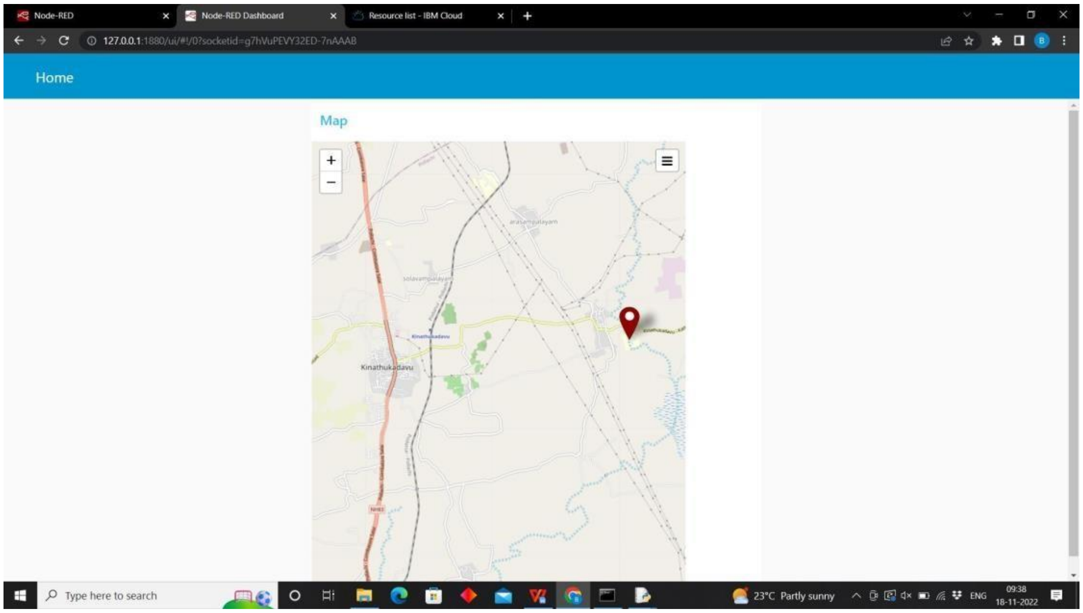
›	□	1234	Disconnected	IoT_device	Device	21 Oct 2022 10:05 AM
›	□	789456	Disconnected	Ultrasonic	Device	29 Oct 2022 2:38 PM
⌄	□	987654321	Connected	gps	Device	17 Nov 2022 10:20 AM

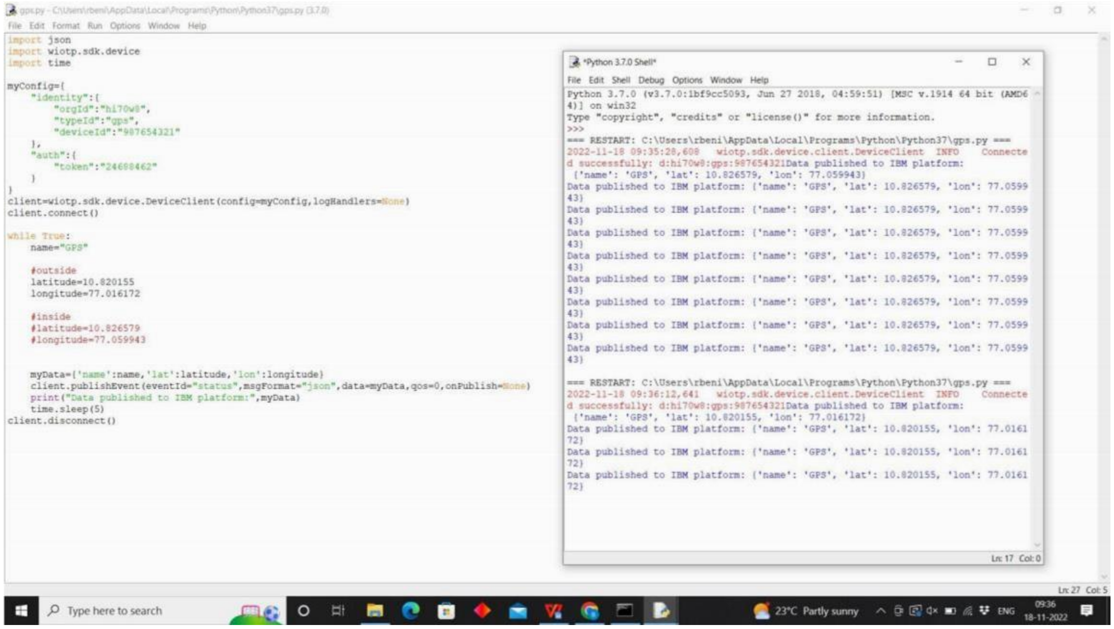
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"name":"GPS","lat":10.820155,"lon":77.016172}	json	a few seconds ago
status	{"name":"GPS","lat":10.820155,"lon":77.016172}	json	a few seconds ago
status	{"name":"GPS","lat":10.826579,"lon":77.059943}	json	a few seconds ago
status	{"name":"GPS","lat":10.826579,"lon":77.059943}	json	a few seconds ago
status	{"name":"GPS","lat":10.826579,"lon":77.059943}	json	a few seconds ago



```
11/18/2022, 9:35:29 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ name: "GPS", lat: 10.826579, lon:
77.059943 }
11/18/2022, 9:35:34 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ name: "GPS", lat: 10.826579, lon:
77.059943 }
11/18/2022, 9:35:39 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ name: "GPS", lat: 10.826579, lon:
77.059943 }
```

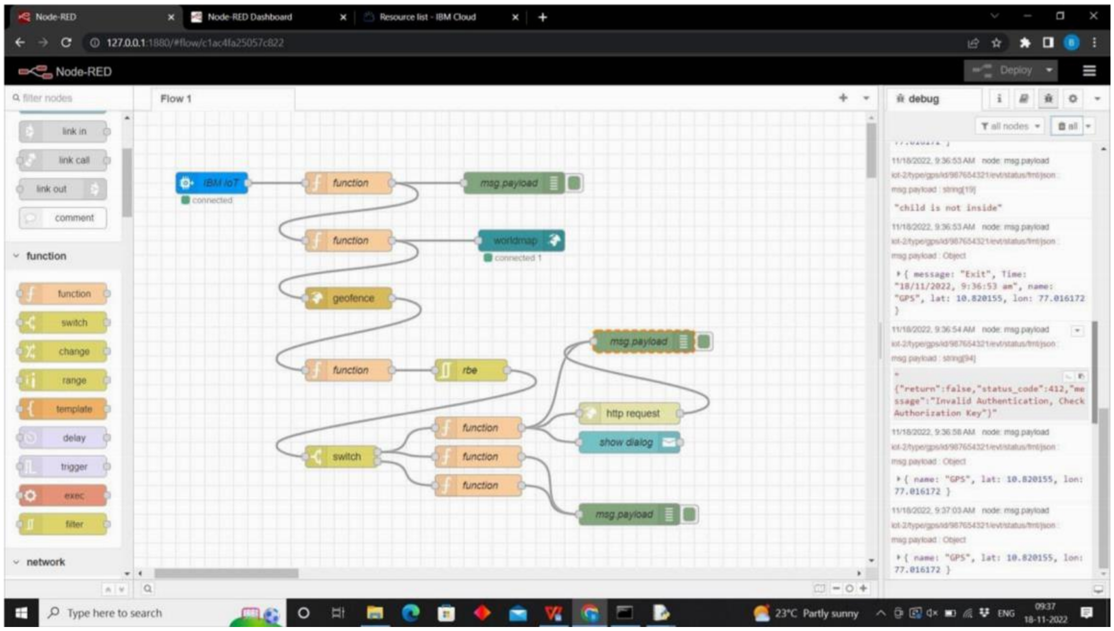




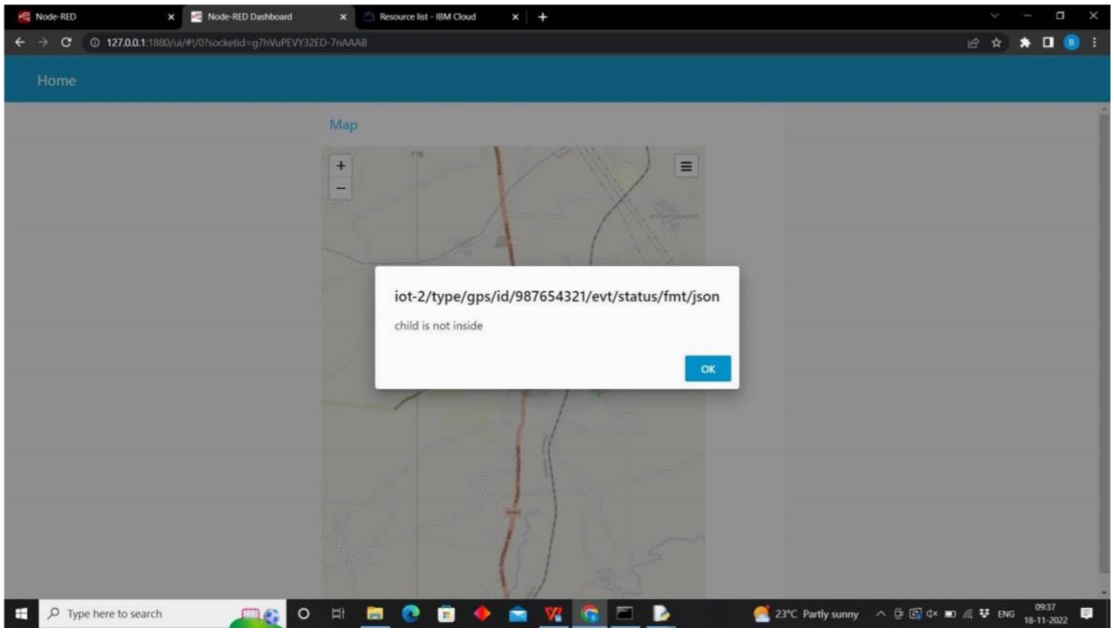
gps.py - C:\Users\rbeni\AppData\Local\Programs\Python\Python37\gps.py (3.7.0)
File Edit Format Run Options Window Help

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

myConfig={
    "identity":{
        "orgId":"hi70w8",
        "typeId":"gps",
        "deviceId":"987654321"
    },
    "auth":{
        "token":"24698462"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()

while True:
    name="GPS"

    #outside
    latitude=10.820155
    longitude=77.016172

    #inside
    #latitude=10.826579
    #longitude=77.059943


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

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\rbeni\AppData\Local\Programs\Python\Python37\gps.py ===
2022-11-18 09:35:28,608   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:hi70w8:gps:987654321Data published to IBM platform:
 {'name': 'GPS', 'lat': 10.826579, 'lon': 77.059943}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.826579, 'lon': 77.0599
43}

=== RESTART: C:\Users\rbeni\AppData\Local\Programs\Python\Python37\gps.py ===
2022-11-18 09:36:12,641   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:hi70w8:gps:987654321Data published to IBM platform:
 {'name': 'GPS', 'lat': 10.820155, 'lon': 77.016172}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.820155, 'lon': 77.0161
72}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.820155, 'lon': 77.0161
72}
Data published to IBM platform: {'name': 'GPS', 'lat': 10.820155, 'lon': 77.0161
72}
```

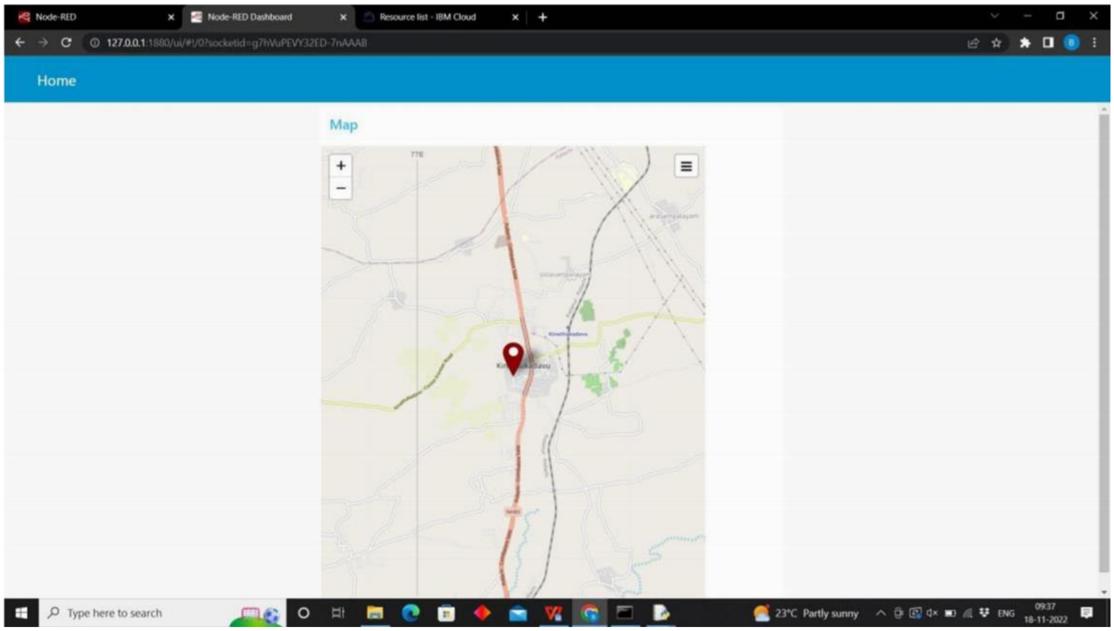


```
11/18/2022, 9:36:53 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : string[19]
"child is not inside"

11/18/2022, 9:36:53 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ message: "Exit", Time:
"18/11/2022, 9:36:53 am", name:
"GPS", lat: 10.820155, lon: 77.016172
}

11/18/2022, 9:36:54 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : string[94]
"
{"return":false,"status_code":412,"me
ssage":"Invalid Authentication, Check
Authorization Key"}"

11/18/2022, 9:36:58 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ name: "GPS", lat: 10.820155, lon:
77.016172 }

11/18/2022, 9:37:03 AM   node: msg.payload
iot-2/type/gps/id/987654321/evt/status/fmt/json :
msg.payload : Object
▸{ name: "GPS", lat: 10.820155, lon:
77.016172 }
```

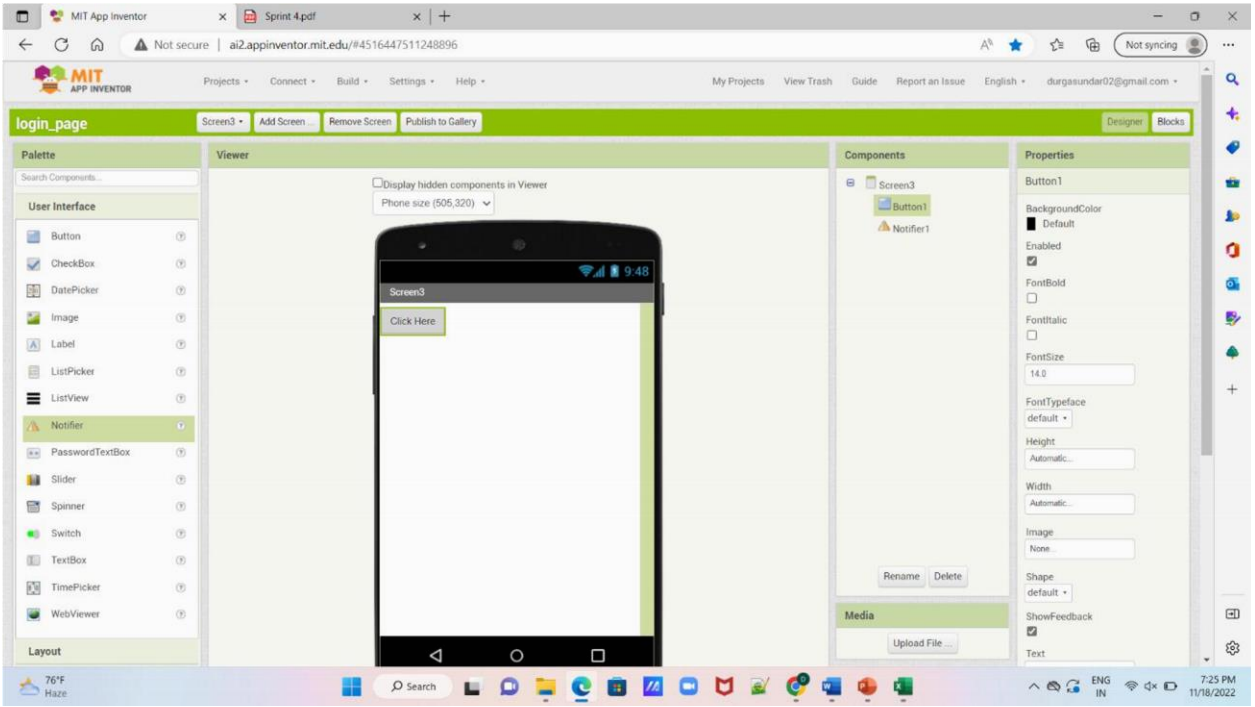


iot-2/type/gps/id/987654321/evt/status/fmt/json

child is not inside



7.4 SPRINT 4:



Display hidden components in Viewer
Phone size (505,320)

Button1

BackgroundColor: Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default

Height: Automatic...

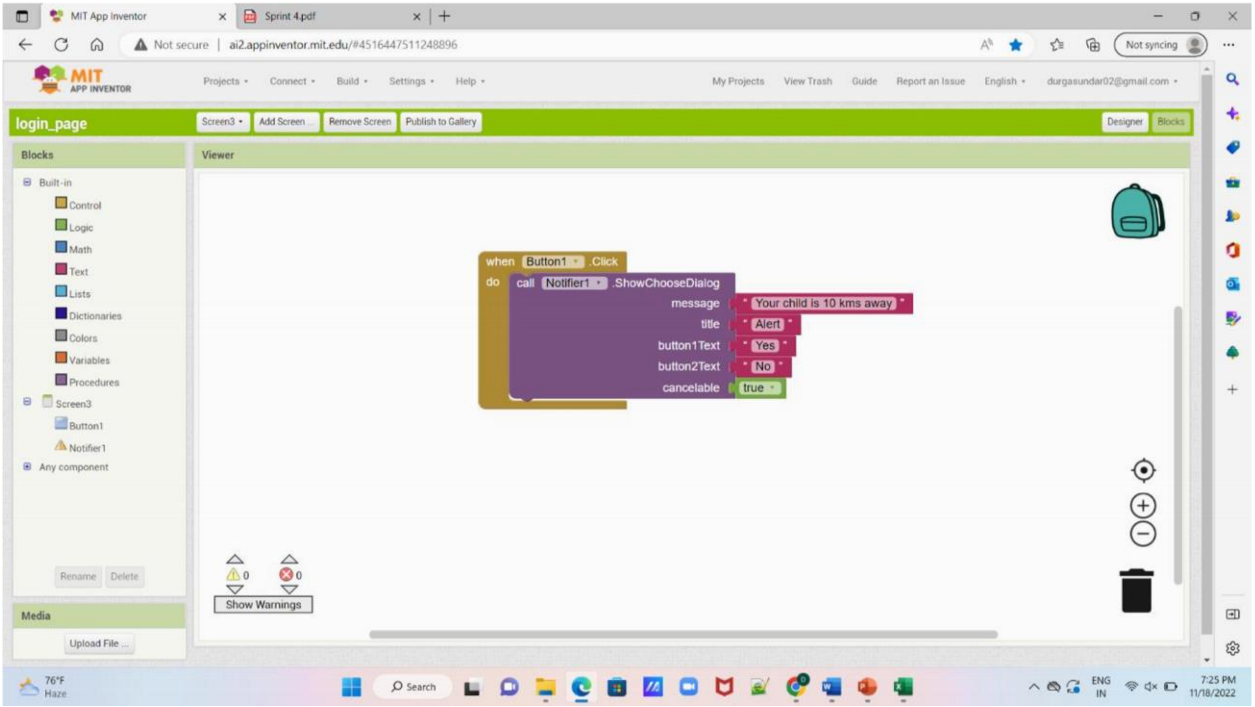
Width: Automatic...

Image: None...

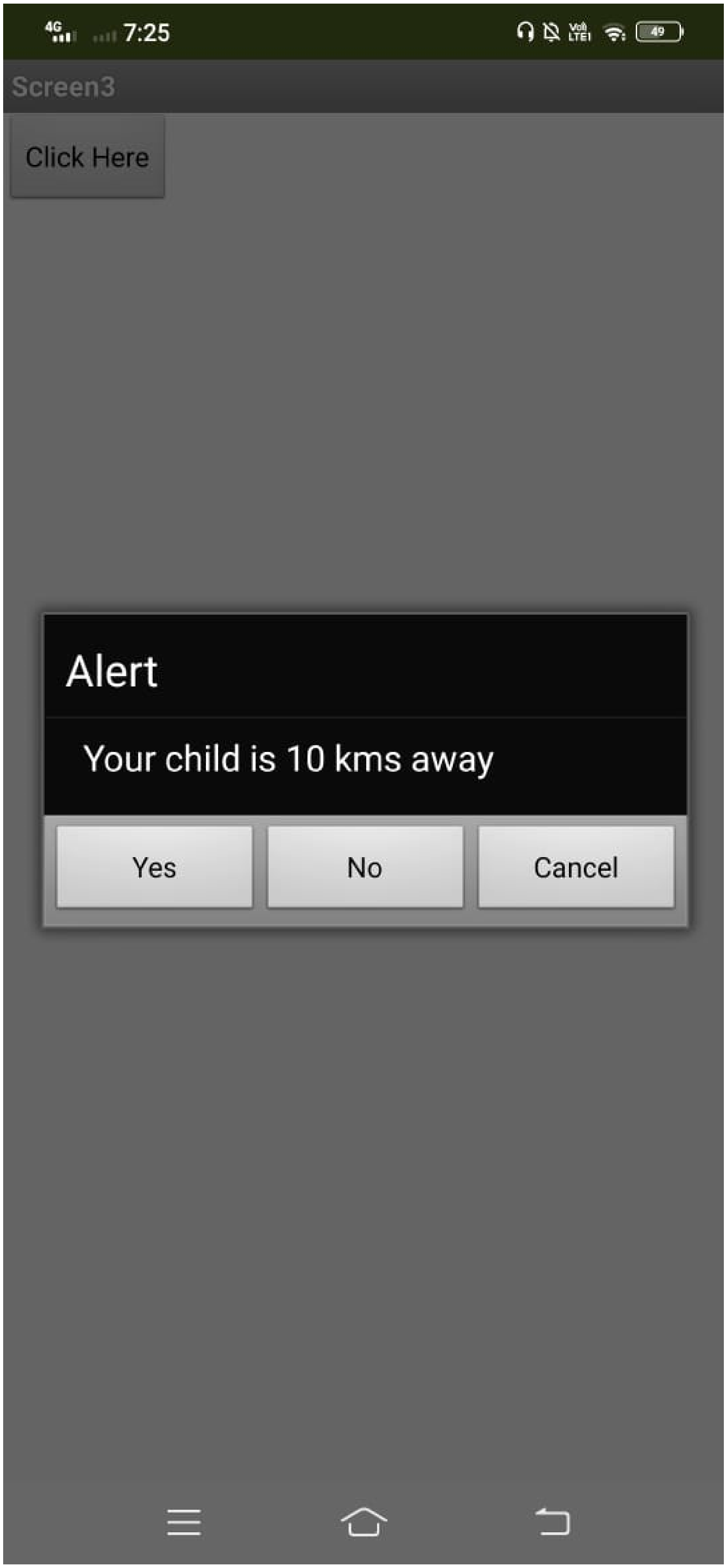
Shape: default

ShowFeedback: ☑

Text



```
when Button1 .Click
do  call Notifier1 .ShowChooseDialog
                          message  " Your child is 10 kms away "
                            title  " Alert "
                      button1Text  " Yes "
                      button2Text  " No "
                       cancelable  true
```



Alert

Your child is 10 kms away

8. TESTING

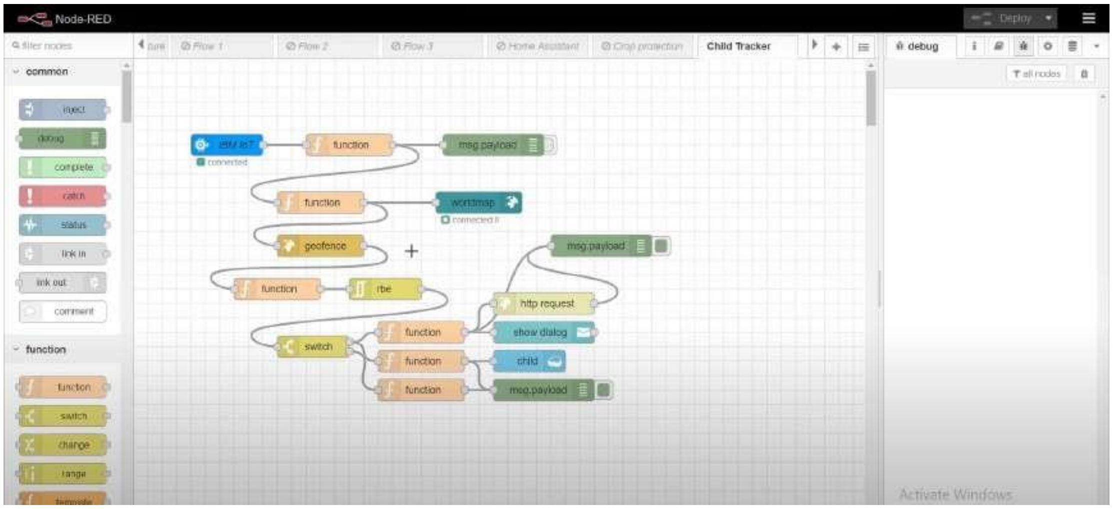
Thus all test cases are tested successfully and the user acceptance testing is also done

9. RESULTS

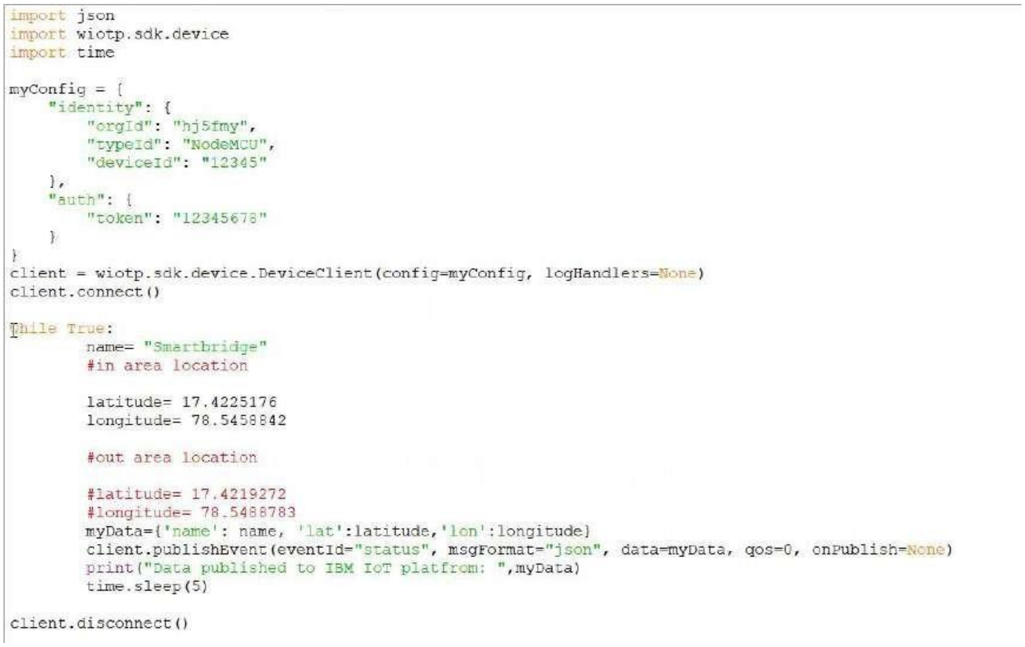
9.1 PERFORMANCE METRICS

Steps :

1. Open a Node-RED project

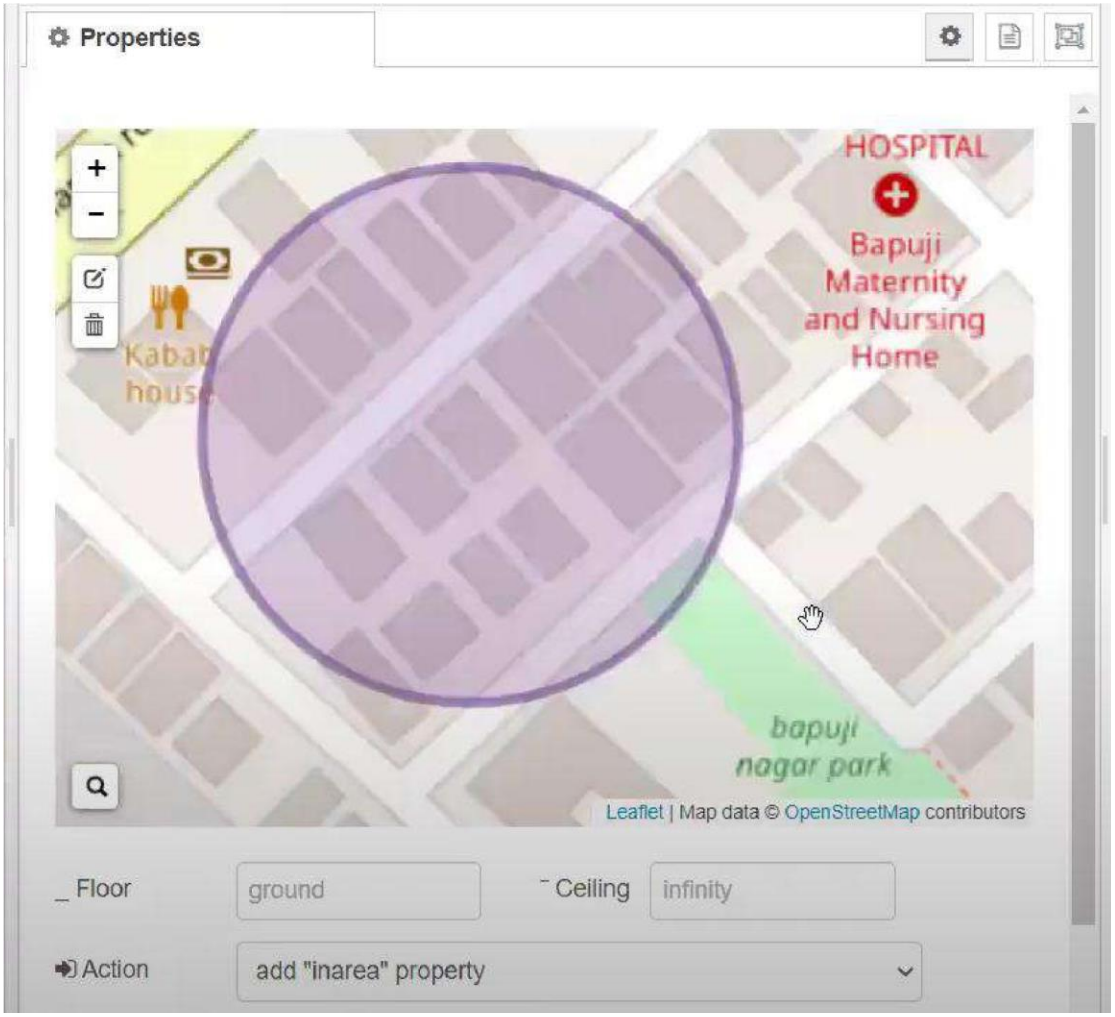


2. Add code to get child location in python



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

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

3. Create the GeoFence

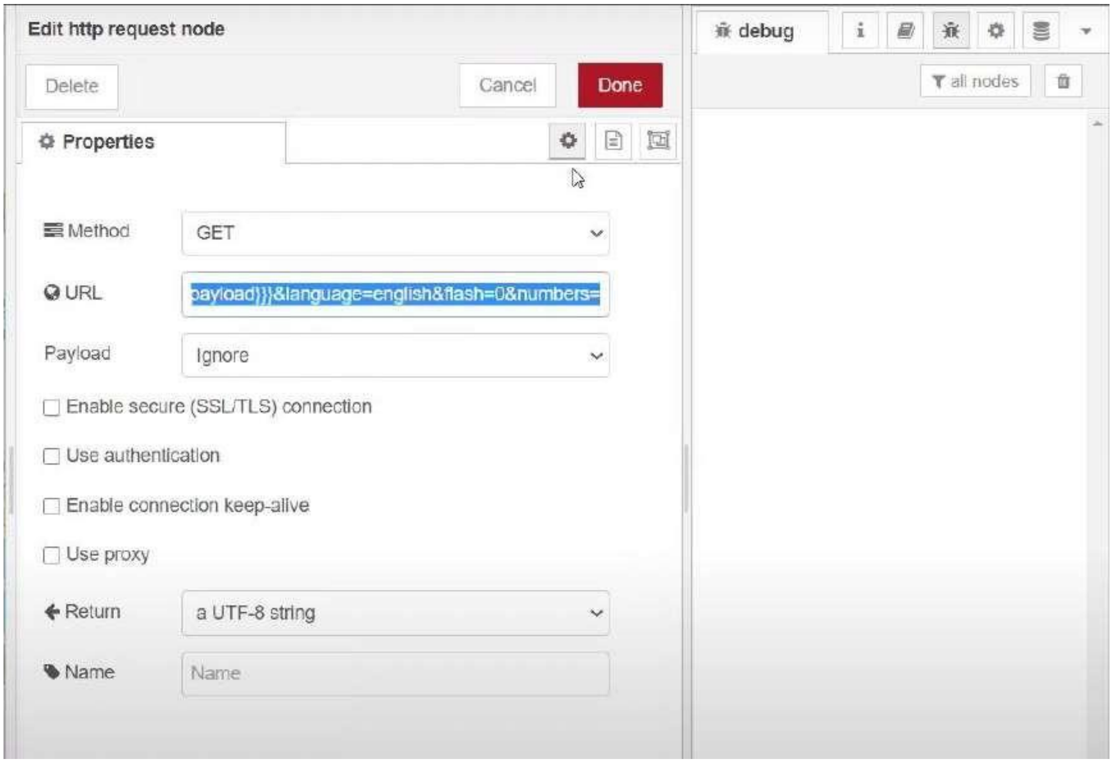


Floor: ground

Ceiling: infinity

Action: add "inarea" property

4. Edit the HTTP Request URL



Method: GET

URL: payload}}}&language=english&flash=0&numbers=

Payload: Ignore

□ Enable secure (SSL/TLS) connection

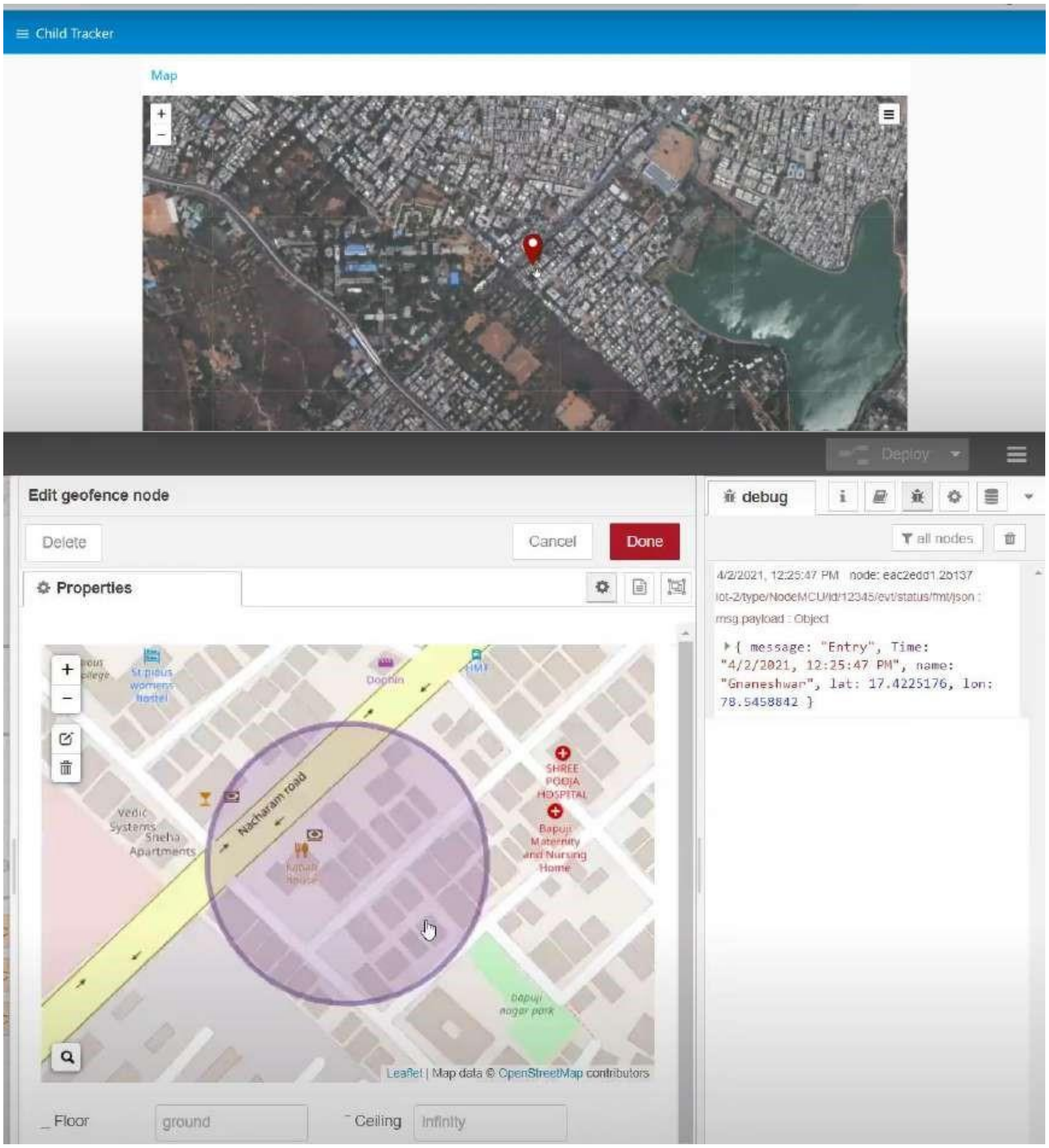
□ Use authentication

□ Enable connection keep-alive

□ Use proxy

Return: a UTF-8 string

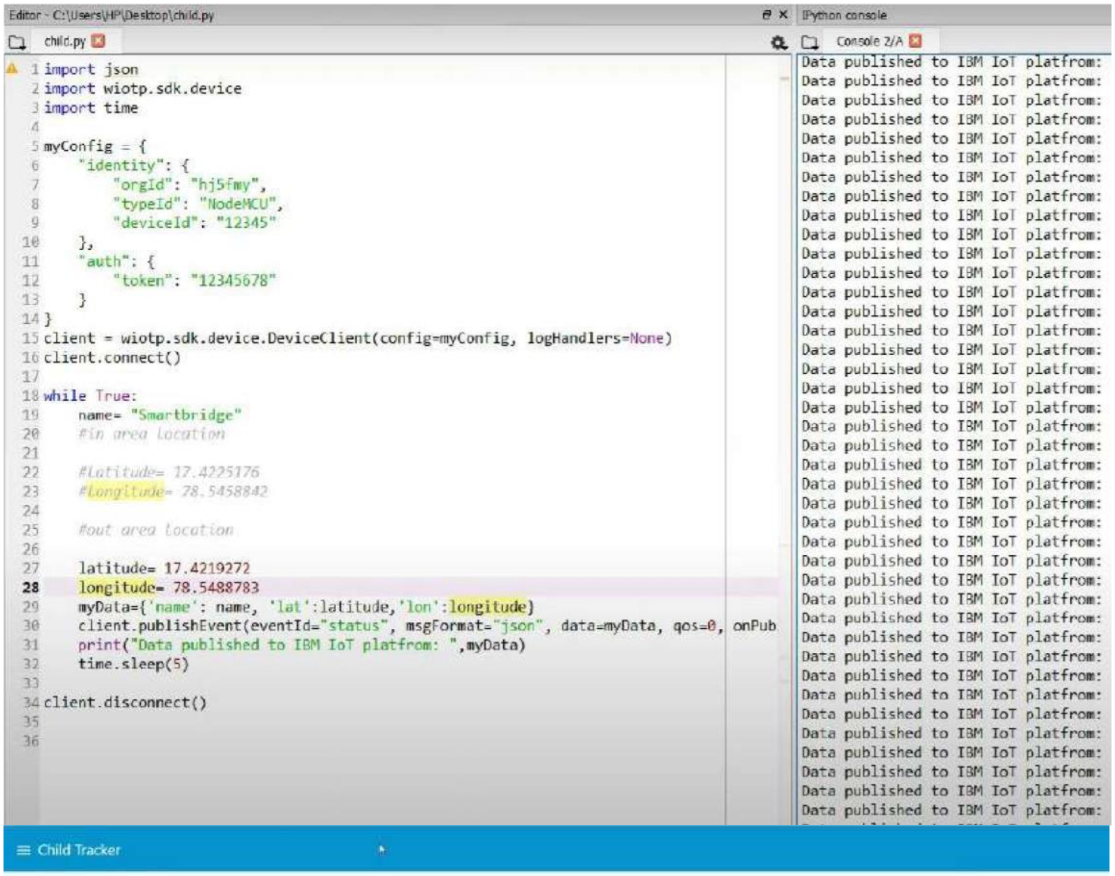
Name: Name



```
4/2/2021, 12:25:47 PM   node: eac2edd1.2b137
iot-2/type/NodeMCU/id/12345/evt/status/fmt/json :
msg.payload : Object
▸{ message: "Entry", Time:
"4/2/2021, 12:25:47 PM", name:
"Gnaneshwar", lat: 17.4225176, lon:
78.5458842 }
```

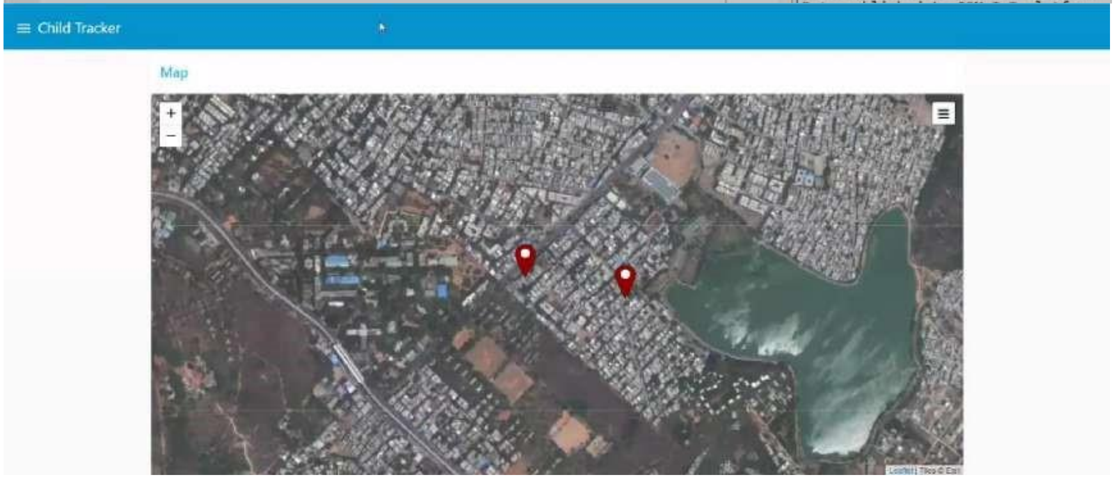
Floor: ground
Ceiling: Infinity

5. Locate the child
6. Create the geofence node
7. Python script send requests to IBM Cloud



```
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, onPub
    print("Data published to IBM IoT platfrom: ",myData)
    time.sleep(5)

client.disconnect()
```

```
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
Data published to IBM IoT platfrom:
```



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

Developed the web application using Node-RED Successfully

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- Ensures safety
- Automatic braking system
- Preventing the stroller being run-away
- Immediate results
- Real-Time visibility on performance
- Automate monitoring
- Automatic alerts and notifications
- Better forensic analysis
- Alert users on time
- Real-Time feedback
- Increased productivity
- Comes with plenty of features.
- Can be used from childbirth to preschool years.
- Is sturdy and very safe for the child.

10.2 DISADVANTAGES

- The use of electronics devices in strollers has not been properly implemented due to the bulkiness of the battery and the clutter mess of wires
- If the stroller is in motion and the battery used for the system reaches a critical level, the solenoid will lose power and extend the metallic rod
- This is the same as applying the brake and such an event could inadvertently cause the stroller to flip over
- Expensive process: Monitoring the child is not an easy process, it is time consuming and costly
- Stressful atmosphere

- Cost in Time and Resources.
- Bulky, feels like moving around furniture.
- The stroller might get worn out by the time it reaches toddlerhood and it is not advisable to use a worn out stroller.
- Assembling can be complex, since there are a lot of parts. Also, moving through narrow doors and spaces would be a problem

11. CONCLUSION

- It has been determined that the architecture of the system is designed to provide parents with security for their children.

- This book is meant for parents and children. Both must possess a modern cell phone that supports GPS and SMS as a foundation. On all modern cell phones, SMS is a core feature, while GPS is only available on new PDAs. The majority of the time, parents will use this tool to locate their children and monitor their mobile device usage to spot inappropriate behavior.

- The application is used to track the child's location as well as their cell phone's call history, messages, and contacts.

- The decision to use the Android OS was made in order to appeal to more customers. The parents will see the children reading history.

12. FUTURE SCOPE

- Last but not least, there is still room for improvement, just like with any product component or setup.
- To improve the framework, features like Geofencing, crisis alerts, and many others can be implemented.
- This technique can be made more effective by adding the IMEI number argument. The suggested system will be put into use, carried on, reviewed, and improved in a subsequent study.