

GRT INSITITUTE OF ENGINEERING AND TECHNOLOGY
–TIRUTTANI – 631209 Affiliated

IoT Based Safety Gadget For Child Safety
Monitoring & Notification

Team Member

- 1.Madhuchaithanya**
- 2.pavankumar**
- 3.Mariyammal**
- 4.Abirose**

AREA OF THE PROJECT

INTERNET OF THINGS(IOT)

OBJECTIVES

- ❖ Enables tracking of the child's location and capturing of data remotely such as where the child located, distance, etc.
- ❖ To show the child's actual data with reference values.
- ❖ Enables sending of notification if the child is out of location or when the device realizes abnormal conditions/situations.
- ❖ Develop a prototype of IoT wearable smart band connected to parent's mobile apps so that they can monitor the actual condition of children at anytime and anyplace.

LITERATURE SURVEY

Author	Title	Proposed Methods	Journal, Year
David Hanes, Gonzalo, Patrick Grosetete, Robert, Barton, Jerome.	“IoT Fundamental and Networking Technologies, Protocols”	During an emergency, mobile apps alert the control room of nearby police stations or caretakers of children. The literature shows that location tracking devices are available in the market but it does not provide a complete solution to the problem. The solution to this problem is to design an IoT device, which senses the child’s location and environment and during an emergency, it should send the alert to the parents automatically.	Cisco,2017
Aditi Gupta, Vibhor Harit.	Child Safety & Tracking Management System by using GPS.	This paper proposed a model for child safety through smartphones that provide the option to track the location of their children as well as in case of emergency children are able to send a quick message and its current location via Short Message Services. Merits: The advantages of smart phones they offer rich features like Google maps, GPS, SMS etc. Demerits: This system is unable to sense the human behaviour of children.	IEEE,2016
K. N. H. Srinivas, T. D. S. Sarveswara Rao, E. Kusuma Kumari.	Smart IoT Device for Child Safety and Tracking.	The system is developed using Link-It ONE board programmed in embedded C and interfaced with temperature, heartbeat, touch sensors and also GPS, GSM & digital camera modules. The novelty of the work is that the system automatically alerts the parent/caretaker by sending SMS when immediate attention is required for the child during an emergency.	IEEE,2019

METHODOLOGY

- ✓ It focuses on the key aspect that a missing child can be assisted by the people around the child and can play a remarkable role in the child's safety until reunited with the parents.
- ✓ If any deviant readings are disclosed by the sensor, then an SMS and phone calls are set off to the parent's mobile. Also, it overhauls the parental app through the cloud.
- ✓ The technique is equipped with GSM and GPS modules for sending and receiving calls, and SMS between the safety gadget and the parental phones.
- ✓ The system also consists of a Wi-Fi/cellular data module used to implement IoT and send 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. Most of the wearables available today are focused on providing the location, and activity of the child to the parents.

METHODOLOGY

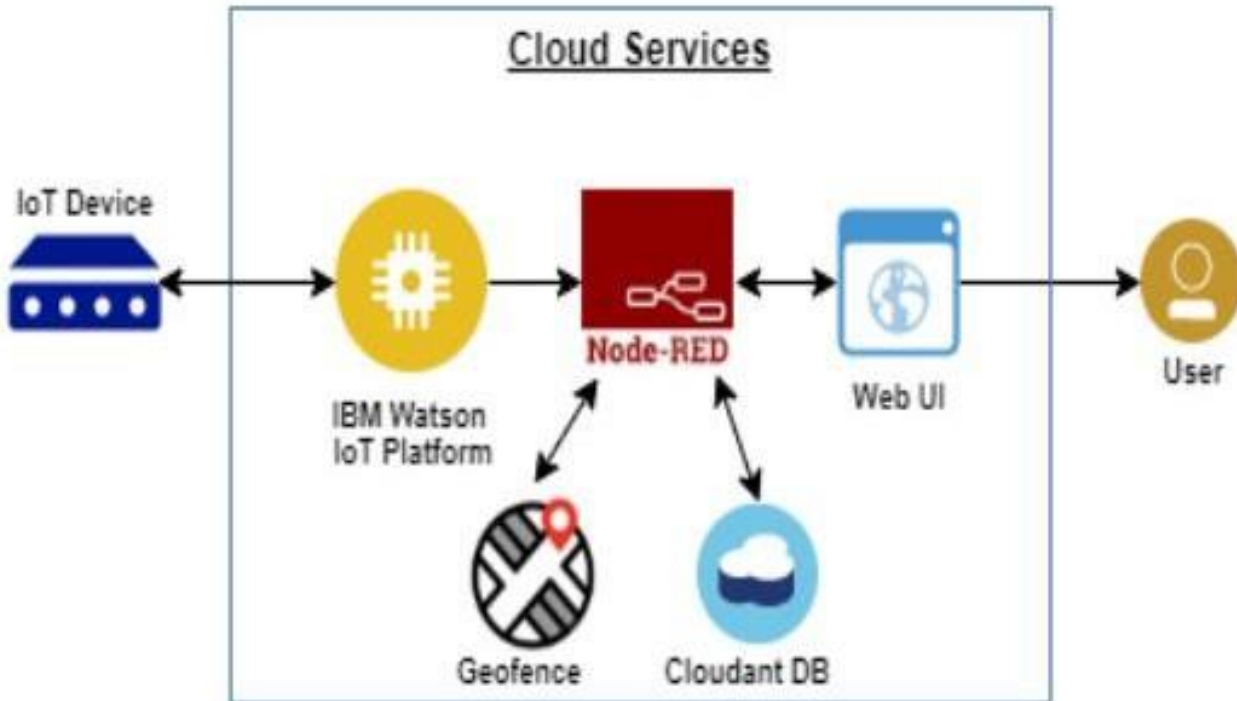
HARDWARE REQUIREMENTS:

IoT deviceGSM
Mobile(Notification)GPS

SOFTWARE REQUIREMENTS:

IBM Cloud
IBM IoT PlatformIBM Node
red IBM Cloudant DB

PROPOSED BLOCK DIAGRAM/TECHNICAL ARCHITECTURE



SIMULATION AND RESULTS

IBM WATSON IOT PLATFORM

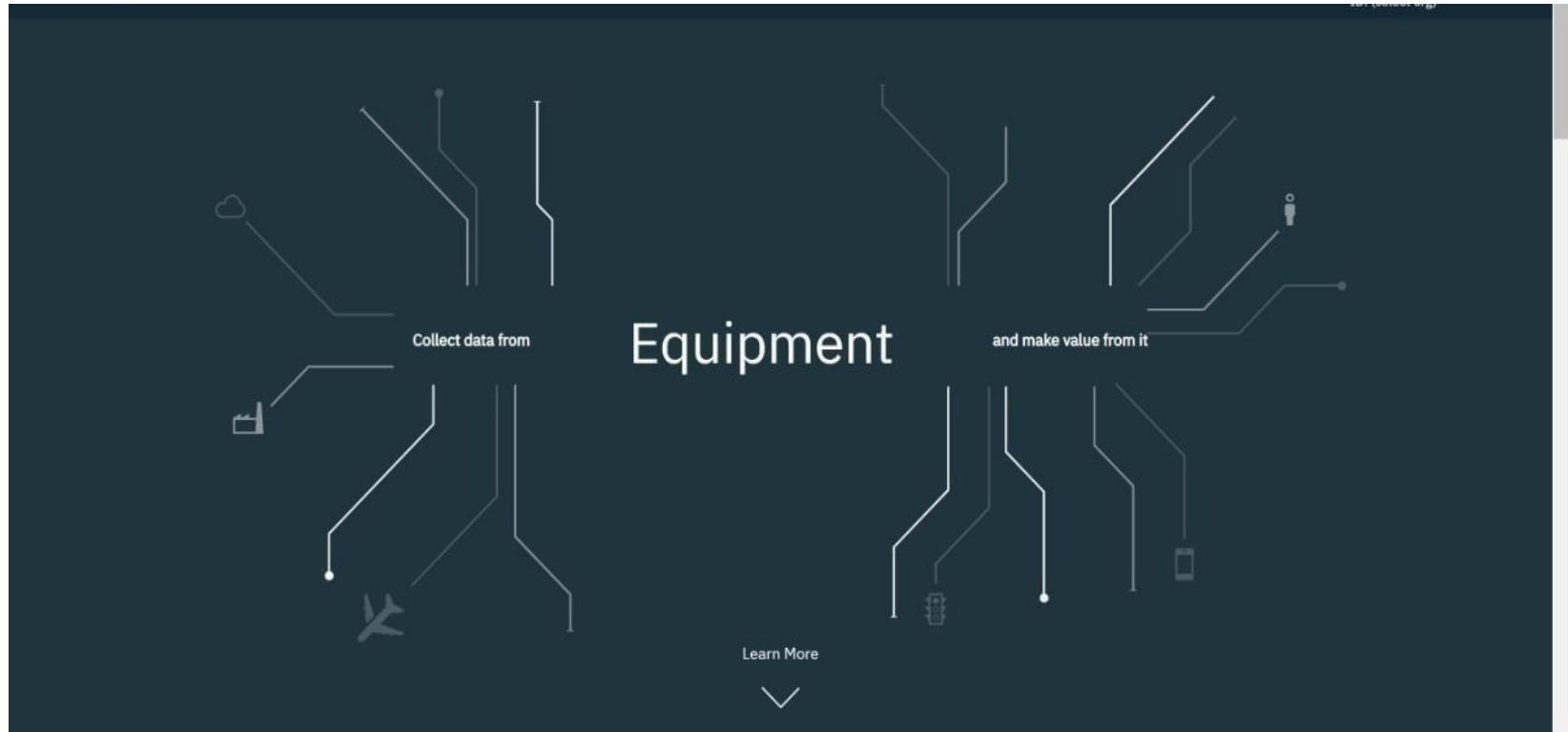


Fig.1,Creating IBM Cloud Service and creating the device

Creating Python Code:

```
import json
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity":{"orgId": "4o1qxb",
    "typeId": "TestDeviceType",
    "deviceId": "12345"},
    "auth": {"token":"pnhXvzN-
sWMKv&hxyi"} }
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()
```




Browse Action Device Types Interfaces

Add Device +

Identity Device Information Recent Events State Logs

X

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":"Smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
status	{"name":"Smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 of 1 page

< 1 >

1 Simulation running

NODE-RED SERVICE

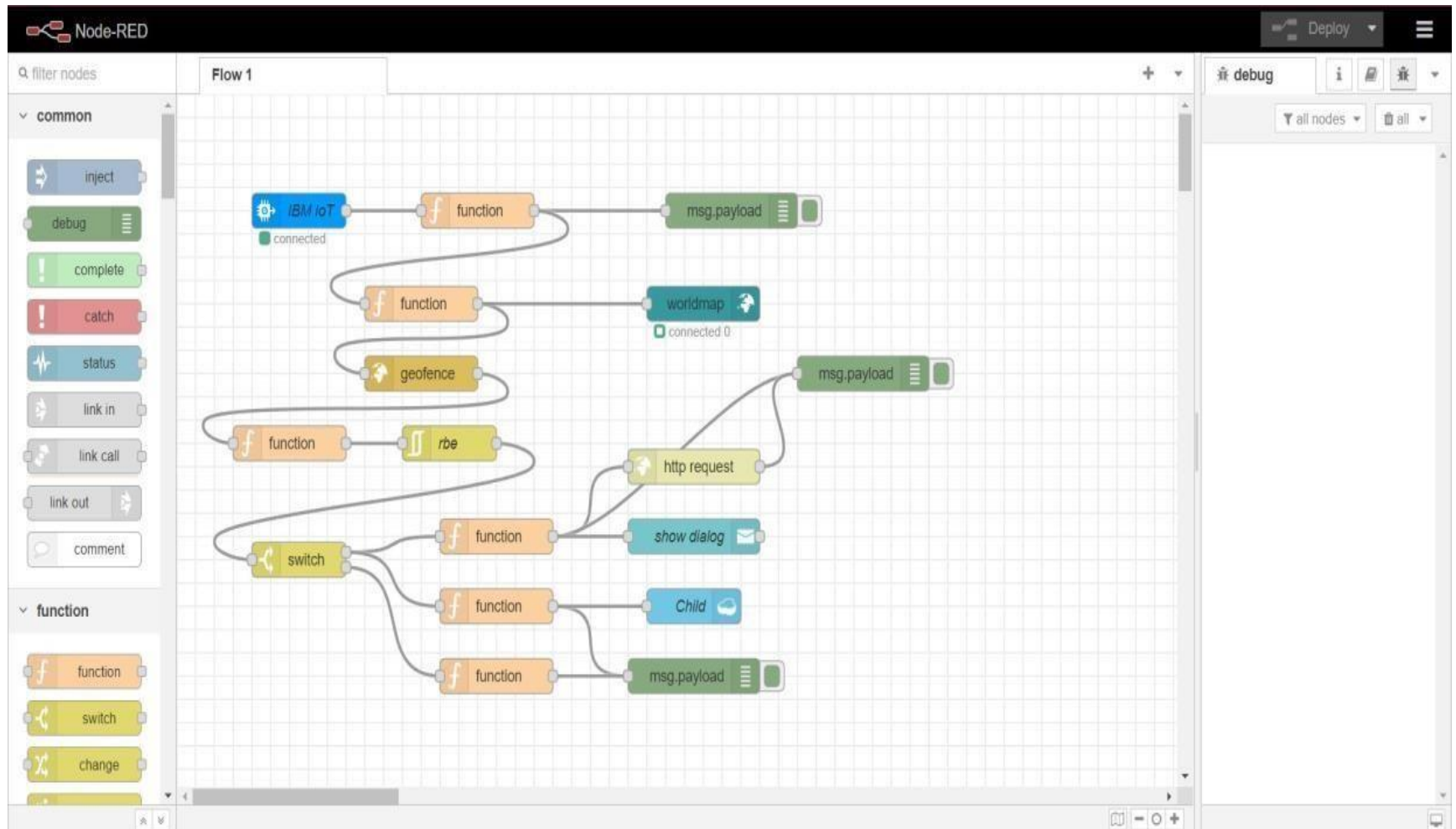


Fig.4,Create Node red service

The image shows the Node-RED web interface. On the left, the 'common' and 'function' node palettes are visible. The main workspace displays 'Flow 1' with a complex flow diagram. The flow starts with an 'IBM IoT' node (status: connected), which connects to a 'function' node. This 'function' node connects to a 'msg.payload' node. The flow then branches into two paths. The top path goes through another 'function' node to a 'worldmap' node (status: connected 0). The bottom path goes through a 'geofence' node to an 'http request' node. The 'http request' node connects to a 'show dialog' node and a 'Child' node. A 'switch' node is also present, which branches into three 'function' nodes. One 'function' node connects to the 'http request' node, another to the 'show dialog' node, and the third to a 'msg.payload' node. The right panel shows the 'Edit ibmiot in node' configuration. The 'Properties' tab is active, displaying various settings for the 'ibmiot' node.

Edit ibmiot in node

Delete Cancel Done

Properties

- Authentication: API Key
- API Key: 813db9c71da271da
- Input Type: Device Event
- Device Type: ☐ All or TestDeviceType
- Device Id: ☐ All or 12345
- 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

Fig.5,Code in each nodes

Node-RED

filter nodes Child Tracker Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

Child Tracker

connected

function

function

geofence

function

Edit function node

Delete Cancel Done

Properties

Name

Setup On Start On Message On Stop

```
1 var name = msg.payload.name
2 var lat = msg.payload.lat
3 var lon = msg.payload.lon
4 global.set('latitude',lat)
5 global.set('longitude',lon)
6 global.set('name',name)
7 return msg;
```

Enabled

dashboard

Layout Site Theme

Tabs & Links

- Child Tracker
- Map

Node-RED

filter nodes Child Tracker Flow 1 Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

IBM IoT

connected

function

msg.payload

function

worldmap

connected 0

geofence

function

rbe

switch

Edit debug node

Delete Cancel Done

Properties

Output

msg. payload

To

- ☒ debug window
- ☐ system console
- ☐ node status (32 characters)

Name

Name

Enabled

dashboard

Layout Site Theme

Tabs & Links

- Child Tracker
 - Map

The screenshot displays the Node-RED web interface. On the left, the 'common' and 'function' node palettes are visible. The main workspace shows a flow titled 'Child Tracker' with a sequence of nodes: 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. A 'function' node is selected, and the 'Edit function node' dialog is open. The dialog shows the 'Properties' tab with a 'Name' field. The 'On Message' tab is active, displaying a JavaScript function that sets the message payload to an object with 'name', 'lat', and 'lon' properties. The right sidebar shows the 'dashboard' tab with a 'Child Tracker' tab and a 'Map' link. The URL at the bottom is <https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>.

Node-RED

filter nodes

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

range

IBM IoT

connected

function

function

worldmap

connected 0

geofence

function

rbe

http request

msg.payload

switch

function

show dialog

function

Child

function

msg.payload

Edit worldmap node

Delete

Cancel

Done

Properties

Group

[Child Tracker] Map

Size

auto

Start

Latitude

Longitude

Zoom

17.4226372

78.5456505

16

Map list

7 selected

Base map

ESRI Satellite

Overlays

5 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

Node-RED

Deploy

filter nodes

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

range

IBM IoT

connected

function

function

function

geofence

function

rbe

switch

function

function

function

function

msg.payload

worldmap

connected 0

http request

show dialog

Child

msg.payload

Edit geofence node

Delete

Cancel

Done

Properties

Map

St. Thomas Cantonment

Guindy National Park

Indian Institute of Technology Madras

Taramani

Thiruvannamiyur

Besant Nagar

Adyar

Kottivakkam

Palavakkam

Neelankarai

Perungudi

Kandanchavai

Madipakkam

Puzhuthivakkam

Ullagaram

Adambakkam

allur

pettai

talai

unnambu

SH109

SH49A

SH10C

Chennai District

Chennai

Leaflet | Map data © OpenStreetMap contributors

_ Floor

ground

_ Ceiling

infinity

Action

add "inarea" property

Enable output of zones to WorldMap node

Enabled

21

Node-RED

filter nodes

Child Tracker

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

function

function

geofence

function

IBM IoT

connected

Edit function node

Delete

Cancel

Done

Properties

Name

Setup

On Start

On Message

On Stop

1 msg.payload=msg.location.inarea

2 return msg;

Enabled

dashboard

Layout

Site

Theme

Tabs & Links

Child Tracker

Map

<https://node-red-opzsk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED interface showing a flow editor and the "Edit filter node" configuration panel.

Flow Editor:

- Flow 1: **Child Tracker** (connected) → **function** → **msg.payload** → **function** → **worldmap** (connected 0) → **geofence** → **function** → **rbe** → **switch**.

Edit filter node configuration:

- Mode:** block unless value changes
- Property:** msg.payload
- ☒ Apply mode separately for each
- Property:** msg.topic
- Name:** rbe
- ☐ Enabled

Dashboard:

- Layout: Site Theme
- Child Tracker
 - Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

filter nodes Child Tracker Flow 1 Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

Child Tracker

Flow 1

Flow 2

IBM IoT

connected

function

msg.payload

function

worldmap

connected

geofence

function

rbe

switch

Edit switch node

Delete Cancel Done

Properties

Name

Property

msg. payload

is false

1

is true

2

+ add

checking all rules

☐ recreate message sequences

Enabled

dashboard

Layout Site Theme

Tabs & Links

- Child Tracker
 - Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/#editor-tab-properties>

Node-RED

filter nodes Child Tracker Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

function

msg payload

workmap

connected 0

geofence

rbe

switch

Edit function node

Delete Cancel Done

Properties

Name

Setup On Start On Message On Stop

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC*3, in your case (UTC*1)
6
7 newDate = new Date(utc + (3600000 * offset));
8
9 msg.payload = {
10   "message": "Exit",
11   "Time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```

Enabled

dashboard

Layout Site Theme

Child Tracker

Map

https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties

Node-RED

Deploy

filter nodes

Child Tracker

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

function

msg.payload

function

worldmap

connected 0

geofence

function

rbe

switch

Edit function node

Delete

Cancel

Done

Properties

Name

Setup

On Start

On Message

On Stop

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)
6
7 newDate = new Date(utc + (3600000* offset));
8
9 msg.payload={
10   "message": "Entry",
11   "Time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```

Enabled

dashboard

Layout

Site

Theme

Tabs & Links

Child Tracker

Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

Deploy

filter nodes: Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

change

range

IBM IoT

connected

function

msg.payload

function

worldmap

connected 0

geofence

function

rbe

http request

msg.payload

switch

function

show dialog

function

Child

function

msg.payload

Edit http request node

Delete Cancel Done

Properties

Method GET

URL <https://www.fast2sms.com/dev/bulkV2?authorizati>

Payload Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

Return a UTF-8 string

Name Name

Enabled

The screenshot displays the Node-RED editor interface. On the left, the 'common' and 'function' node palettes are visible. The main workspace shows a flow diagram with nodes including 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', 'comment', 'function', 'msg.payload', 'worldmap', 'geofence', 'rbe', and 'switch'. The 'Edit notification node' configuration panel is open on the right, showing settings for the 'OK / Cancel Dialog' layout. The 'Send to all browser sessions' checkbox is checked. The 'Default action label' is set to 'OK'. The 'Secondary action label' is set to '(optional label for Cancel button)'. The 'Accept raw HTML/JavaScript input in msg.payload to format popup' checkbox is unchecked. The 'Class' is set to '[msg.className]', the 'Topic' is set to '[msg.topic]', and the 'Name' is set to 'Show Dialoge'. A note at the bottom of the panel states: 'Note: checking Accept raw HTML/JavaScript can allow injection of'. The right sidebar shows the 'dashboard' tab selected, with 'Layout', 'Site', and 'Theme' options available. The 'Tabs & Links' section shows a tree view with 'Child Tracker' and 'Map' tabs.

Node-RED

Deploy

filter nodes

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

range

IBM IoT

connected

function

msg.payload

function

worldmap

connected 0

geofence

function

rbe

http request

msg.payload

switch

function

show dialog

function

Child

function

msg.payload

Edit cloudant out node

Delete

Cancel

Done

Properties

Service: External cloudant or couchdb service

Server: https://93dc87d4-df75-4ee1-a851-2026

Database: childsafelydb

Operation: insert

☐ Only store msg.payload object?

Name: Child

☐ Enabled

Connecting with IBM Cloud: Using IBM IOT nodethrough the API key

Browse IBM Cloud Apps

+ Generate API Key

Browse API Keys

Type the app description to search for

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

<input type="checkbox"/>	Key	Description	Role	Expires	
2 results					
<input type="checkbox"/>	a-4o1qxb-d5wguvebrf	-	Standard Application	-	<div></div>
<input type="checkbox"/>	a-4o1qxb-ecmygwzdce	API Key for the device simulator	Standard Application	-	<div></div>

1 Simulation running

Apps using your microphone: Google Chrome



Browse IBM Cloud Apps

+ Generate API Key

Browse API Keys

Type the app description to search for



This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).



Key

Description

Role

Expires



2 results



a-4o1qxb-d5wguvebrf

Standard Application

API Key Information

Access Control/Permissions



Key a-4o1qxb-d5wguvebrf

Last Edited By 310819106007@smartinternz.com

Description

-

Expires

Never

Date Added Nov 10, 2022 2:20 PM

Last Update Nov 10, 2022 2:20 PM

1 Simulation running

Transferring values from Python Code:

```
child.py - C:\Users\Anu\AppData\Local\Programs\Python\Python37\chld.py (3.7.0)
File Edit Format Run Options Window Help
import json
import wiotp.sdk.device
import time
myConfig = {
    "identity":{
        "orgId": "4o1qxb",
        "typeId": "TestDeviceType",
        "deviceId": "12345"
    },
    "auth": {
        "token": "pnHxvzN-sWMMkvshxyi"
    }
}
client= wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

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

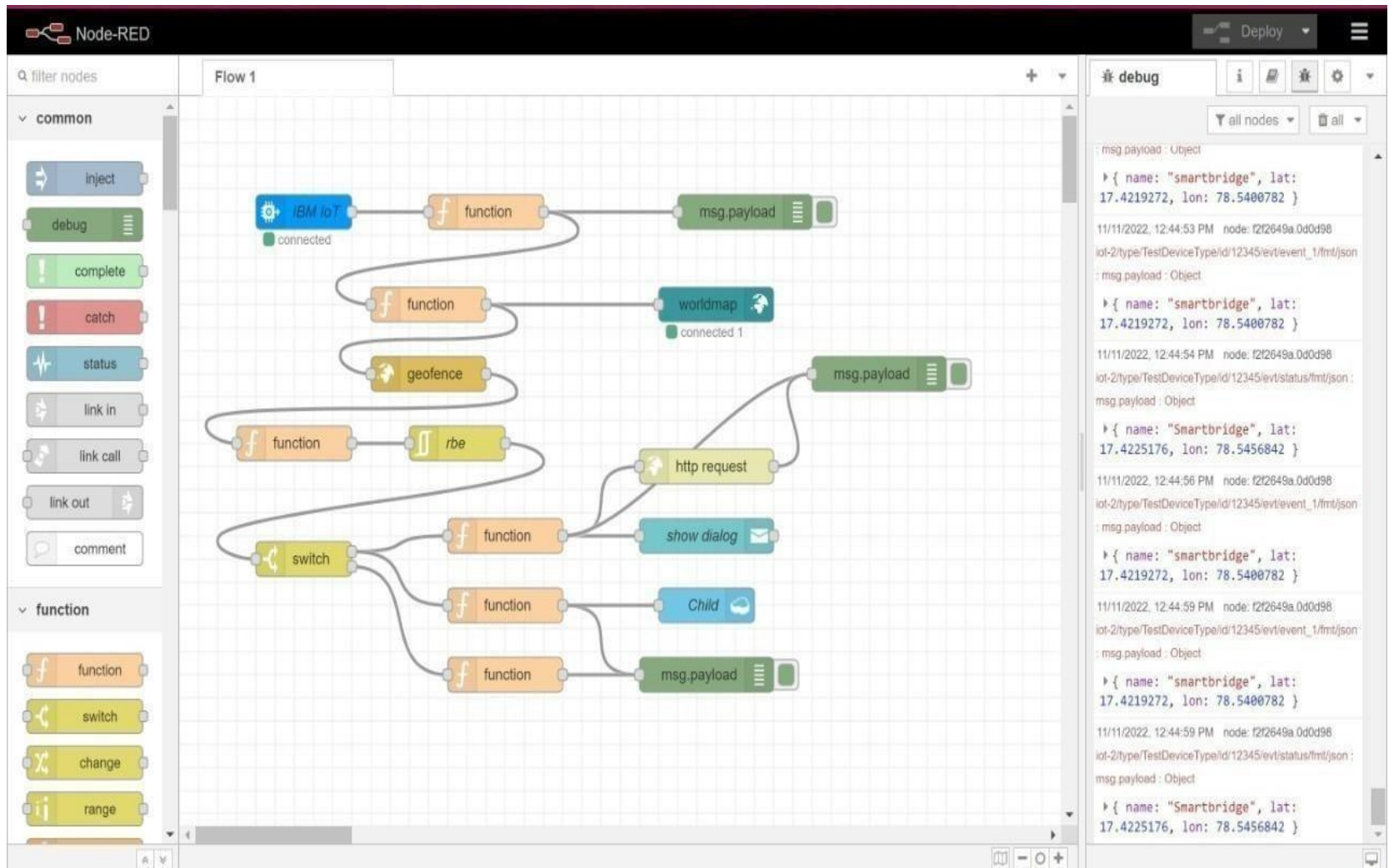
    #latitude = 17.4225176
    #longitude = 78.5456842

    #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 platform: ",myData)
    time.sleep(5)







client.disconnect()
```


Node-Red:



Creating Cloudant DB and integrating Node-Red with the Web UI

The screenshot displays the Cloudant Databases web interface. On the left is a dark sidebar with navigation icons. The main header area includes the title 'Databases', a 'Database name' dropdown, a 'Create Database' button, and links for JSON, documents, and notifications. Below the header, the section 'Your Databases' contains a table with two entries:

Name	Size	# of Docs	Partitioned	Actions
childsafety	14 bytes	1	No	  
childsafetydb	15 bytes	1	No	  

At the bottom of the interface, a status bar indicates 'Showing 1-2 of 2 databases.' and 'Databases per page 20'. A pagination control shows '1' as the current page.

child safety db
Document ID
Options
{ } JSON

All Documents +

Query

Permissions

Changes

Design Documents +

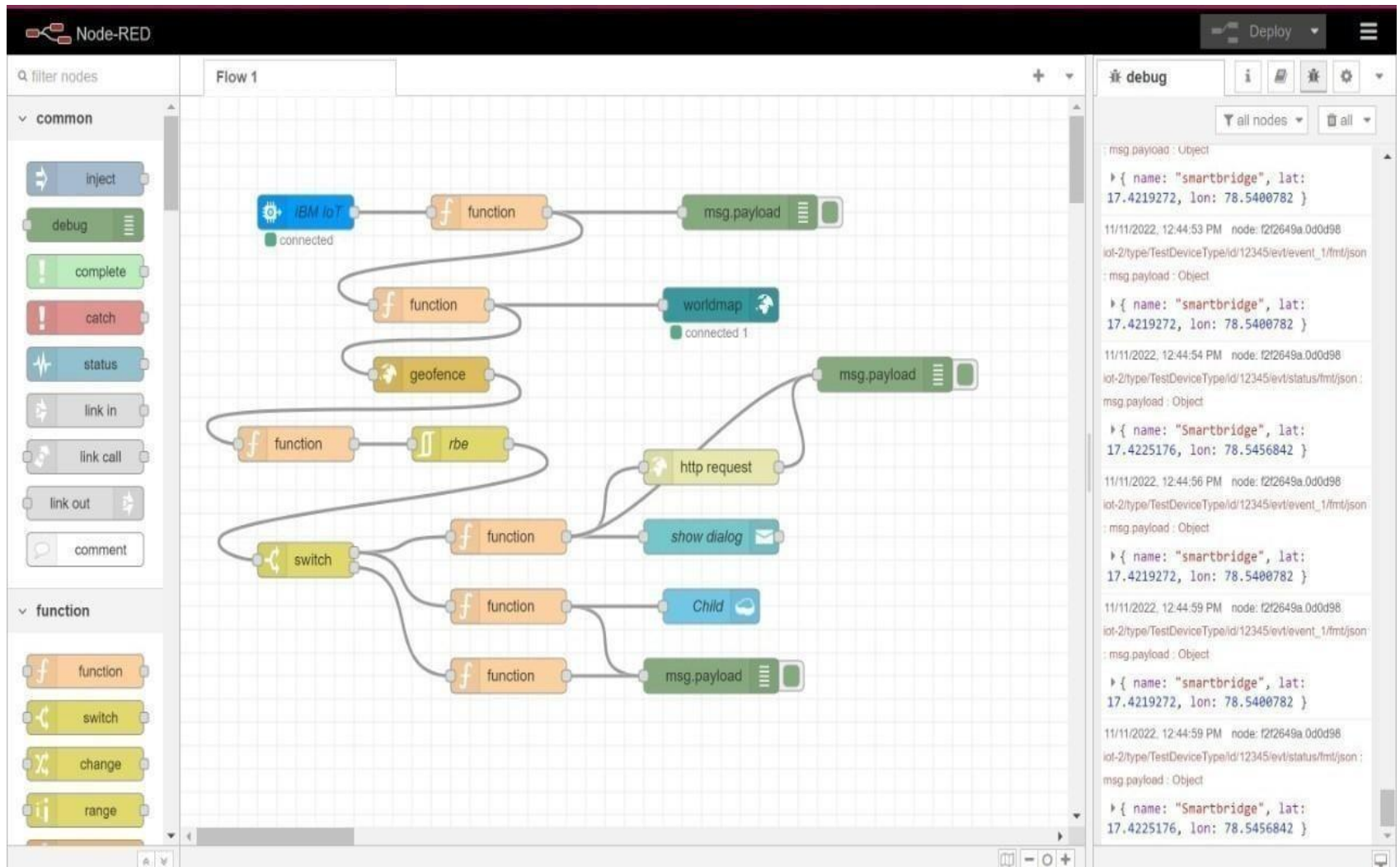
☐

Table	Metadata	{ } JSON
Create Document		

	id	key	value
<input type="checkbox"/>	e375b703e3d589d1a5e0108d5c7fd2...	e375b703e3d589d1a5e0108d5c7fd2...	{"rev": "2-3f59b999d6fe5a17091eba..."}

Showing document 1 - 1.
Documents per page: 20
< >

Node-Red Service with Cloudant Database:



Node-RED

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

IBM IoT (connected)

function

msg.payload

function

worldmap (connected 0)

geofence

function

rbe

http request

msg.payload

switch

function

show dialog

function

Child

function

msg.payload

Edit cloudant out node

Delete Cancel Done

Properties

Service: External cloudant or couchdb service

Server: https://93dc87d4-df75-4ee1-a851-2026

Database: childssafetydb

Operation: insert

☐ Only store msg.payload object?

Name: Child

Enabled

Node-Red Dashboard(Web ui):

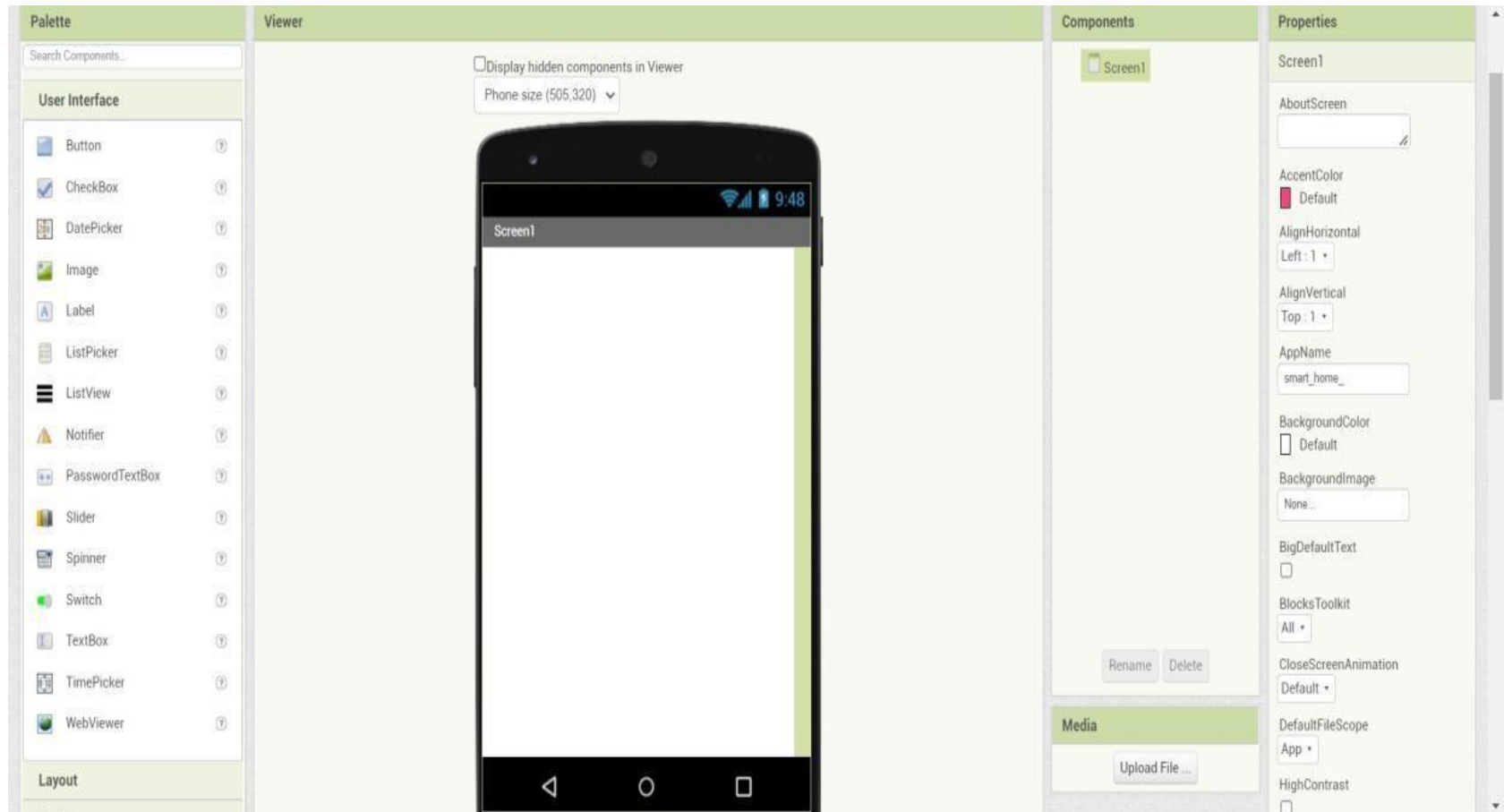
Child Tracker

Map



Creating the MIT app and Showing the child's location

Create App in MIT App inventor:



Block Configuration:

The screenshot displays the 'ChildTracker' application's block configuration environment. The interface is divided into several sections:

- Top Bar:** Contains the app name 'ChildTracker', a dropdown menu for 'Screen1', and buttons for 'Add Screen ...', 'Remove Screen', and 'Publish to Gallery'. On the right, there are tabs for 'Designer' and 'Blocks'.
- Left Panel (Blocks):** A hierarchical tree view showing the app's structure. It includes 'Screen1' with a 'VerticalArrangement1' containing 'TextBox1', 'Label1', 'Button1', 'Label2', 'Label3', 'Label4' (highlighted), and 'Button2'. Below this is a 'Map1' component with 'Circle1', 'Marker1', and 'Marker2'. Further down are 'LocationSensor1', 'FirebaseDB1', and 'Navigation1'. At the bottom of this panel is an 'Any component' section with 'Rename' and 'Delete' buttons.
- Bottom Left (Media):** A section with an 'Upload File ...' button.
- Center (Viewer):** The main workspace for configuring blocks. It shows a sequence of blocks:
 - 'Initialize global flag to 0'.
 - 'when FirebaseDB1 .DataChanged' block with a 'tag' dropdown set to 'value'.
 - A 'do' block containing:
 - 'if' block: 'get value' equals 'lat', then 'set Label3 . Text' to 'get value'.
 - 'else if' block: 'get value' equals 'long', then 'set Label3 . Text' to 'get value'.
 - 'else if' block: 'get value' equals 'address', then 'set Label2 . Text' to 'get value'.

At the bottom of the viewer, there are two warning icons (a yellow triangle and a red X) with a 'Show Warnings' button. On the right side of the viewer, there is a backpack icon and a vertical toolbar with a target icon, a plus sign, a minus sign, and a trash can icon.

[Privacy Policy and Terms of Use](#)

ChildTracker

Screen1Add Screen ...Remove ScreenPublish to Gallery

DesignerBlocks

Blocks

Screen1

- VerticalArrangement1
 - TextBox1
 - Label1
 - Button1
 - Label2
 - Label3
 - Label4
 - Button2
- Map1
 - Circle1
 - Marker1
 - Marker2
- LocationSensor1
- FirebaseDB1
- Navigation1

Any component

RenameDelete

Media

Upload File ...

Viewer

when Button2 .Click

do

- set Label3 .Text to "17.4219272"
- set Label4 .Text to "78.5488783"
- call Map1 .PanTo
 - latitudeLocationSensor1 .Latitude
 - longitudeLocationSensor1 .Longitude
 - zoom15

when Navigation1 .GoDirections


directions points distance duration




do


- set Navigation1 .StartLocation to Marker1
- set Navigation1 .EndLocation to Marker2
- call Navigation1 .RequestDirections

Show Warnings

00







Privacy Policy and Terms of Use

Blocks

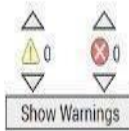
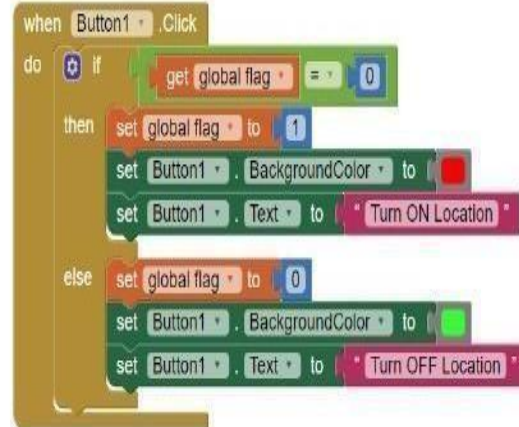
- Screen1
 - VerticalArrangement1
 - TextBox1
 - Label1
 - Button1
 - Label2
 - Label3
 - Label4
 - Button2
 - Map1
 - Circle1
 - Marker1
 - Marker2
 - LocationSensor1
 - FirebaseDB1
 - Navigation1
- Any component

Rename Delete

Media

Upload File ...

Viewer



ChildTracker Screen1 Add Screen... Remove Screen Publish to Gallery Designer Blocks

Blocks

- Screen1
 - VerticalArrangement1
 - TextBox1
 - Label1
 - Button1
 - Label2
 - Label3
 - Label4
 - Button2
 - Map1
 - Circle1
 - Marker1
 - Marker2
 - LocationSensor1
 - FirebaseDB1
 - Navigation1
- Any component

Rename Delete

Media

Upload File...

Viewer

when LocationSensor1 .LocationChanged

latitude longitude altitude speed

do set Label1 .Text to "Updated"

if get global flag = 0

then

call FirebaseDB1 .StoreValue

tag "lat"

valueToStore get latitude

call FirebaseDB1 .StoreValue

tag "long"

valueToStore get longitude

call FirebaseDB1 .StoreValue

tag "address"

valueToStore LocationSensor1 .CurrentAddress

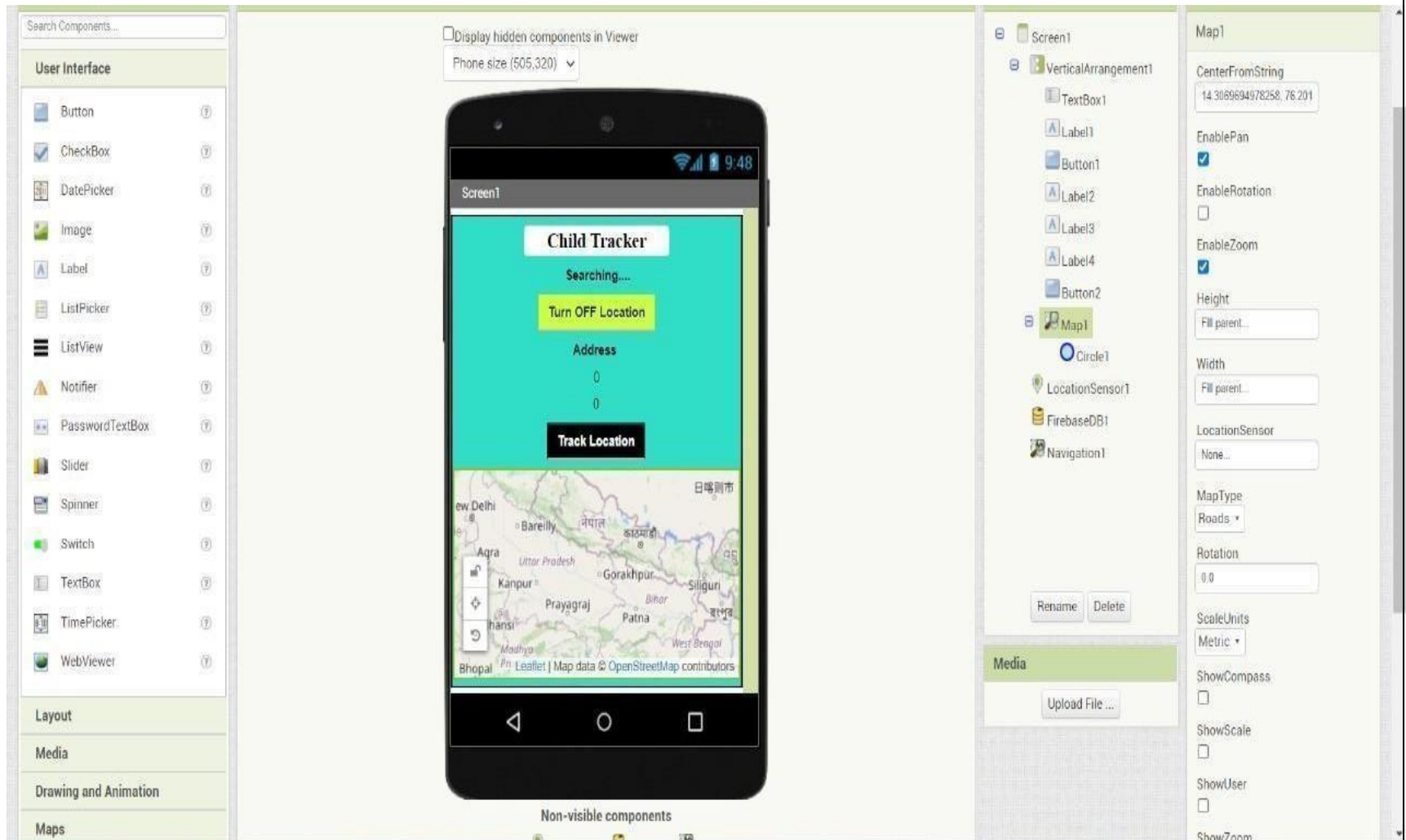
set latitude to "17.4219272"

set longitude to "78.5488783"

Show Warnings

Privacy Policy and Terms of Use

Output(App inventor):



Location Status:



CONCLUSION

This paper to ensure the safety of children and increase their confidence. Many experimenters are operating in this area and have formulated different technologies to aid children. The key represented in this paper takes the advantage of smartphones which proposes affluent elements like Google maps, SMS, etc. The child safety and protection device is proficient in acting as a smart IoT device. It equips parents with real-time location, the surrounding temperature, and along with an alarm buzzer for their child's circumstances and the capability to locate their child. This paper depicts the fundamental design concept and functionality along with the anticipated consequences.

REFERENCES

- {1} Authors: David Hanes, Gonzalo, Patrick Grosetete, Robert, Barton, Jerome. Title: Henry "IoT Fundamental and Networking Technologies, Protocols"- CISCO 2016
https://books.google.co.in/books/about/IoT_Fundamentals.html?id=F6GxjgEACAAJ&redir_esc=y
- {2} Authors: Aditi Gupta, Vibhor Harit. Published in: 2016 IEEE. Title: ChildSafety & Tracking Management System by using GPS
<https://scholar.archive.org/work/djydnxvovbdhbbthlunfw7tye>
- {3} Authors: K. N. H. Srinivas, T. D. S. Sarveswara Rao, E. Kusuma Kumari. Title: Smart IoT Device for Child Safety and Tracking. Published in: 2019 IEEE.
<https://ijsrcseit.com/paper/CSEIT206288.pdf>
- {4} Authors: Akash Moodbidri, Hamid Shahnasser. Title: Child safety wearable device. Published in: 2017 IEEE. <https://ieeexplore.ieee.org/document/7899531>